

Chapter 19

Epilogue: The Same Algorithms Give the Same Results

©2005 by Harvey Gould, Jan Tobochnik, and Wolfgang Christian
4 April 2005

We emphasize that the methods we have discussed can be applied to a wide variety of natural phenomena.

19.1 The Unity of Physics

Although we have discussed many topics and applications, we have covered only a small fraction of the possible computer simulations and models of natural phenomena. However, we know that the same physical principles can be applied to many kinds of phenomena. We express this point of view as *the same algorithms give the same results*. For example, the Monte Carlo methods that we applied to the simulation of classical liquids and to the analysis of quantum mechanical wave functions also were applied to the transport of neutrons and problems in chemical kinetics. Similar Monte Carlo methods are being used to analyze problems in quark confinement. Indeed, the increasing role of the computer in research is strengthening the interconnections of the various subfields of physics and the relation of physics to other disciplines.

The computer also has helped us think of natural phenomena in new ways that complement traditional methods. For example, consider a predator-prey model of the dynamics of minnows and sharks. Assume that the birth rate of the minnows is independent of the number of sharks, and that each shark kills a number of minnows proportional to their number. If we assume that $F(t)$, the number of minnows at time t , changes continuously, we can write

$$\frac{dF(t)}{dt} = [b_1 - d_1 S(t)]F(t), \quad (19.1)$$

where $S(t)$ is the number of sharks at time t , and b_1 and d_1 are constants independent of F and S . To obtain an equation for the rate of change of the sharks, we assume that the number of offspring

produced by each shark is proportional to the number of minnows eaten by the shark. If we also assume that the death rate of the sharks is constant, we can write

$$\frac{dS(t)}{dt} = [b_2 F(t) - d_2] S(t). \quad (19.2)$$

Equations (19.1) and (19.2) are known as the Lotka-Volterra equations. They can be analyzed by standard methods and solved numerically using simple algorithms. Why is the dynamical behavior of (19.1) and (19.2) cyclic?

In the predator-prey model the numbers of predator and prey are assumed to change continuously and their spatial distribution is ignored. We now summarize an alternative model that can be most simply expressed as a computer algorithm. The model is a two-dimensional cellular automaton known as Wa-Tor.

1. For a desired concentration of minnows and sharks, minnows and sharks are placed at random on the sites of a lattice. The minnows and sharks are assigned random ages.
2. At time step t_n , consider each minnow sequentially. Determine the number of nearest neighbor sites that are unoccupied at time t_{n-1} and move the minnow at random to one of the unoccupied sites. If all the nearest neighbor sites are occupied, the minnow does not move.
3. If a minnow has survived for a multiple of `fbreed` iterations, the minnow has a single offspring. The new minnow is placed at the previous position of the parent minnow.
4. At time step t_n , consider each shark sequentially. If all the nearest neighbor sites of the shark at time t_{n-1} are unoccupied, move the shark at random to one of the four unoccupied sites. If one or more of the adjacent sites is occupied by a minnow, the shark moves at random to one of the occupied sites and eats the minnow.
5. If a shark moves `nstarve` times without eating, the shark dies. If a shark survives for a multiple of `sbreed` iterations, the shark has a single offspring. The new shark is placed at the previous position of the parent shark.

What is the dynamical behavior of Wa-Tor? Do Wa-Tor and the Lotka-Volterra equations exhibit similar behavior? Is the Wa-Tor model realistic? What are the advantages and disadvantages of each approach? See the references for suggestions for the numerical values of the parameters.

19.2 Percolation and Galaxies

In addition to allowing us to investigate complex nonlinear problems and more realistic systems, the computer has reinforced one of the contemporary themes in physics, the unifying role of collective behavior. Systems composed of many individual constituents can exhibit common properties under certain conditions, even though there might be differences in the nature of the constituents and in their mutual interaction. The behavior of a system near a critical point is probably the best example of collective behavior in a familiar context. In the following, we discuss examples of collective behavior in the context of epidemiology and the structure of spiral galaxies. Our discussion follows closely the articles by Schulman and Seiden.

Consider an imaginary disease called *percolitis*. The disease conveys no immunity, and its incubation period and duration are both one day. The disease is so benign that its sufferers are able to come into contact with every member of the community so that every person comes into contact with every other person every day. At $t = 0$ one person contracts the disease from a source outside the community. Let N be the total population, t the time measured in days, p the transmission probability, and $n(t)$ the expected number of diseased individuals at time t . Convince yourself that for $N = 1000$ and $p = 0.0005$, the chance that there will be anyone suffering from the disease seven days later is vanishingly small. On the other hand, suppose that $p = 0.002$. Then $n(t = 1) = 2$, $n(t = 2) \approx 4$, and the odds are very high that after some time there will be an average number of approximately 800 victims. Do a simulation for various values of N and p and estimate the critical probability p_c such that for $p < p_c$ the average number of victims is zero, and for $p \geq p_c$ the average number of victims is nonzero. Note that no assumptions were made as to which individuals will become infected.

As its name suggests, the percolitis model has much in common with percolation. What are some of the connections? The model is sufficiently simple that an analytical solution is possible. What modifications of the model might make it more realistic for thinking about the spread of epidemics? Would you have imagined some of these modifications without thinking about how to put the problem on a computer? Do these modifications change the qualitative behavior of the model? Are analytical solutions possible in general?

The internal structure of a galaxy has traditionally been studied using Newtonian dynamics. This point of view is very useful, but can be complemented by thinking about the large scale structure of a galaxy using ideas from statistical mechanics. Because we can only briefly summarize this alternative point of view here, we encourage you to explore the properties of the percolation-based model of Schulman and Seiden by running **GalaxyApp**, a simple version of their model. The basic assumption of the model is that even though a region of the galaxy might have the necessary ingredients for star formation, nothing happens if it is left alone. However, if a shock wave from a supernova passes through the gas, there is a good chance that a star will be formed. The supernova is itself the result of an earlier nearby star formation. The theory of *self-propagating star formation* is based on the importance of this mechanism. We can think of a given region of the galaxy as being like a percolitis-susceptible individual—without a source there is no percolitis. Rather than determining which regions have the necessary conditions for star formation, we summarize all the uncertainty and variability in a single parameter p , the probability that a supernova explosion in one region gives rise to star formation in a neighboring region.

The other important observation we need to make about spiral galaxies is that galaxies do not rotate rigidly (with a constant angular velocity), but to a good approximation each region rotates with the same tangential velocity. The properties of random self-propagating star formation and constant tangential velocity are incorporated into **GalaxyApp** in Listing 19.1 as follows. Imagine dividing a galaxy into concentric rings, which are divided into cells of equal size (see Fig. 19.1). Initially, a small number of cells are activated. Each cell corresponds to a region of space that is the size of a giant molecular cloud and moves with the same tangential velocity v . The angular velocity is given by $\omega = v/r$, where r is the distance of the ring from the center of the galaxy. At each time step, the active cells activate neighboring cells with probability p , and then become inactive. Then the rings are rotated, and the process is repeated again in the next time step. At each time step, cells that have been active within the last 15 time steps are plotted with filled boxes, with the size of each box inversely proportional to the time since the cell become active. More details of the

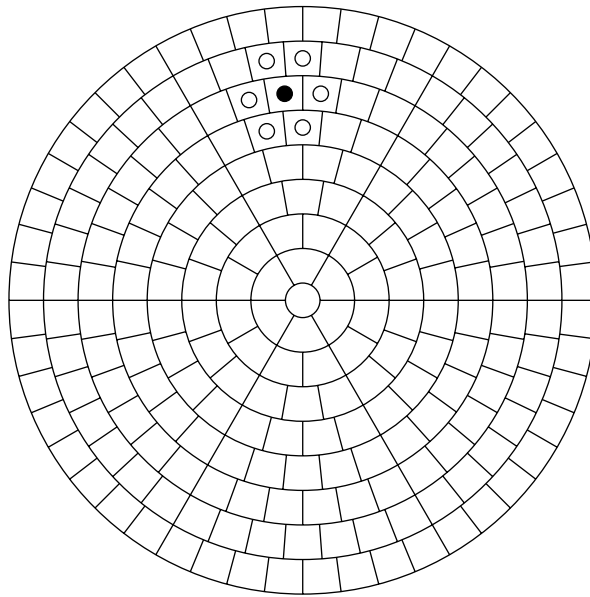


Figure 19.1: The nature of the polar grid used in **Program galaxy**. Each cell has the same area and has on the average six nearest neighbors. The filled circle denotes an active region of star formation. At the next time step it can induce star formation in cells containing open circles. Note that as time passes, the neighbors in adjacent rings change because of differential rotation.

simulation are shown in Fig. 19.1 and in Listing 19.1. A typical galaxy generated by **GalaxyApp** is shown in Fig. 19.2.

Listing 19.1: Galaxy simulation program.

```
package org.opensourcephysics.sip.ch19;

import org.opensourcephysics.controls.*;
import org.opensourcephysics.frames.*;
import org.opensourcephysics.display.*;
import java.awt.*;

public class GalaxyApp extends AbstractSimulation implements Drawable {

    DisplayFrame frame = new DisplayFrame("Galaxy");

    final static double twoPi = 2.0*Math.PI;
```

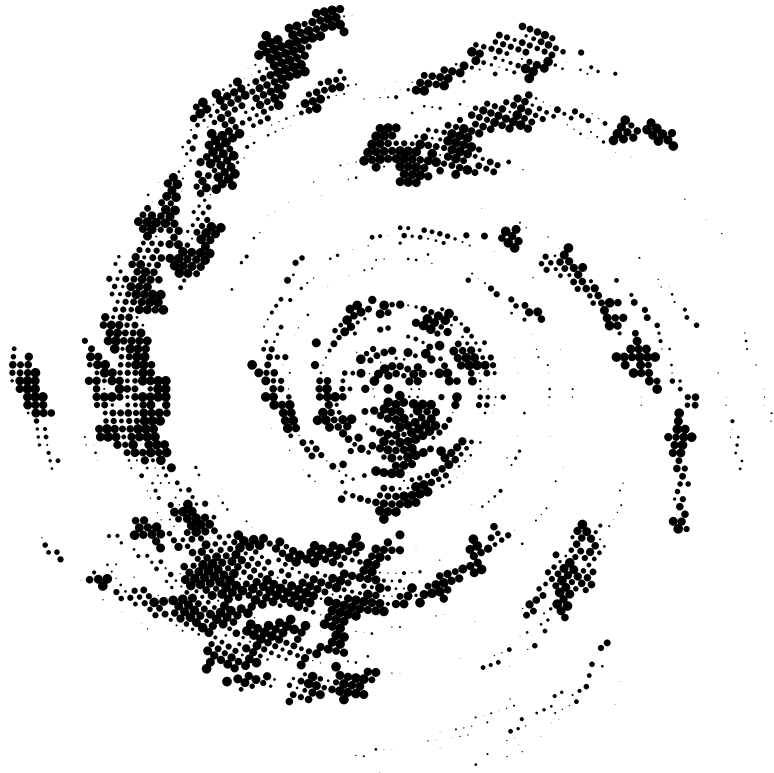


Figure 19.2: A typical structure generated by `GalaxyApp`. The parameters are the number of rings `nring` = 50, the initial number of active cells `nactive` = 200, circular velocity $v = 1$ (200 km/s), the probability of induced star formation $p = 0.18$, and the time step `dt` = 10 (10^7 years). The structure shown is at $t = 200$ with 358 active star clusters. The diameter of the circle representing a star cluster is proportional to the remaining lifetime of the cluster.

```
final static double twoPiOver6 = twoPi/6;

double p, v, dt, t;

int numberOfRings, numberOfActiveCells, numberOfCells;

int [] starLifeTime, cellR, cellA, activeCellLabel;

public GalaxyApp() {
    frame.addDrawable(this);
}
```

```

public void initialize () {

    t = 0;

    numberOfRings = control.getInt("Number of rings");

    numberOfActiveCells = control.getInt("Initial number of active cells");

    p = control.getDouble("star formation probability");

    v = control.getDouble("cell velocity");

    dt = control.getDouble("time step");

    frame.setPreferredMinMax(-numberOfRings-1.0, numberOfRings+1.0, -numberOfRings-1.0, numberOfRings+1.0);

    numberOfCells = 3*numberOfRings*(numberOfRings+1);

    cellR = new int[numberOfCells];

    cellA = new int[numberOfCells];

    int cellLabel = 0;

    // initial values of r and a for each cell label

    for(int r = 1;r<=numberOfRings;r++) {

        for(int a = 0;a<r*6;a++) {

            cellR[cellLabel] = r;

            cellA[cellLabel] = a;

            cellLabel++;

        }

    }

    starLifeTime = new int[numberOfCells];

    activeCellLabel = new int[numberOfCells];

    initializeGalaxy ();

}

```

```

public void initializeGalaxy() {

    int i = 0;

    while(i<numberOfActiveCells) {

        int label = (int) (Math.random()*numberOfCells);

        if(starLifeTime[label]!=15) {

            starLifeTime[label] = 15; // activate region for 15 time steps

            activeCellLabel[i] = label;

            i++;

        }

    }

}

```

```

public void formNewStars() {

    // copy list of active cells into another array

    int [] currentActiveCellLabel = (int []) activeCellLabel.clone();

    int currentNumberOfActiveCells = numberOfActiveCells;

    numberOfActiveCells = 0;

    for(int i = 0; i<currentNumberOfActiveCells; ++i) {

        int cellLabel = currentActiveCellLabel[i];

        int r = cellR[cellLabel];

        int a = cellA[cellLabel];

        // activate neighboring cells in same ring

        createStars(r, pbc(a+1, r));

        createStars(r, pbc(a-1, r));
    }
}

```

```

        //activate cells in next larger ring
        if(r<numberOfRings-1) {
            int ap = aForOtherRadius(a, r, r+1);
            createStars(r+1, pbc(ap, r+1));
            createStars(r+1, pbc(ap+1, r+1));
        }
        //activate cells in next smaller ring
        if(r>1) {
            int am = aForOtherRadius(a, r, r-1);
            createStars(r-1, pbc(am, r-1));
            createStars(r-1, pbc(am+1, r-1));
        }
    }
}

public int pbc(int a, int r) {
    return(a+6*r)%(6*r);
}

public int aForOtherRadius(int a, int r, int rOther) {
    // angle corresponding to a at time t
    double angle = twoPiOver6*a/r+((v*t)/r);
    angle -= twoPi*(int) (angle/twoPi);
    // change in angle for cell at other r
    double angleChange = ((v*t)/rOther);

```

```

    angleChange -= twoPi*(int) (angleChange/twoPi);

    // return value of a for cell at other r
    return(int) ((rOther/twoPiOver6)*(angle-angleChange));
}

public void createStars(int r, int a) {
    int label = a+3*r*(r-1);

    if((Math.random()<p)&&(starLifeTime[label]!=15)) {
        activeCellLabel[numberOfActiveCells] = label;

        numberOfActiveCells++;

        starLifeTime[label] = 15;
    }
}

public void doStep() {
    t += dt;

    formNewStars();

    frame.setMessage("t = "+decimalFormat.format(t)+" #active = "+numberOfActiveCells);
}

public void reset() {
    control.setValue("Number of rings", 50);

    control.setValue("Initial number of active cells", 200);

    control.setValue("star formation probability", 0.18);

    control.setValue("cell velocity", 1.0);
}

```

```

        control.setValue("time step" , 10.0);
    }

    public void draw(DrawingPanel panel, Graphics g) {
        for(int label = 0; label < numberOfCells; label++) {
            if(starLifeTime[label] > 0) {
                int r = cellR[label];
                int a = cellA[label];

                double angle = twoPiOver6*a/r+(v*t)/r; //angle for cell at time t

                double x = r*Math.cos(angle);
                double y = r*Math.sin(angle);

                double ds = starLifeTime[label]/15.0;

                int leftPixels = panel.xToPix(x);
                int topPixels = panel.yToPix(y);

                int widthPixels = panel.xToPix(ds)-panel.xToPix(0);
                int heightPixels = panel.yToPix(0)-panel.yToPix(ds);

                g.fillOval ( leftPixels , topPixels , widthPixels , heightPixels );

                starLifeTime[label]--;           // decrease star cluster lifetime
            }
        }
    }

    public static void main(String[] args) {
        SimulationControl.createApp(new GalaxyApp());
    }

```

```
}

```

Of course our brief discussion of galaxies is not meant to convince you that the mechanism proposed by Schulman and Seiden is correct. Rather our purpose is to show how an alternative point of view can suggest new approaches in different fields. As emphasized by the authors, the dramatic images produced by computer simulations of the galaxy model show unanticipated features that can be the impetus for the development of a deeper theoretical understanding.

19.3 Numbers, Pretty Pictures, and Insight

The power of physics comes in part from its ability to give numerical agreement between theory and experiment. However, numerical agreement has little significance, unless the process of obtaining that agreement leads to insight into the phenomena of interest. For example, it is possible to design elaborate epicycle models of planetary motion which yield numerical results that are consistent with observations. Nevertheless, we prefer the Copernican approach, not for its impressive numerical success, but because it provided insight and lead to further advances by Kepler and Newton.

Computer simulations raise similar questions. The numbers produced by simulations which are consistent with experimental observations, and the pictures that are suggestive of physical phenomena are not sufficient to establish the value of a simulation. As an example, let us briefly consider a simulation of river networks. You might have seen aerial photographs of the Earth's topography and noticed the fractal-like drainage patterns formed by many rivers. A variety of random walk models can be used to generate patterns that look remarkably like river networks and even share some of their statistical properties. In these models the path of a walker represents a river, and the branching and intersections of rivers are modeled by the intersection of the paths of many walkers. However, such models have limited utility because they provide little insight into why the rivers actually have the properties they do. They do not directly incorporate the important physical processes of erosion and sedimentation.

Leheny proposed a model that produces realistic looking patterns of river networks whose statistical properties are consistent with observations of real networks. Unlike the random walk models, the dynamics of this lattice model reflect actual physical processes. The model consists of first creating a terrain for the network, and then defining the network on the terrain. The model can be summarized as follows:

1. The initial terrain is assumed to have a constant slope m . Each site of the lattice is given an initial height, $h(x, y) = my$.
2. Precipitation is placed on a random site of the (square) lattice.
3. Water flows from this site to a nearest neighbor site with a probability proportional to $e^{E\Delta h}$, where Δh is the difference in height between the site and a neighbor, and E is a parameter. If $\Delta h < 0$, then the probability is equal to zero, and the flow will not return to the site previously visited if there is a nonzero probability of flowing to another site.
4. Step (3) is repeated until the water flows to the bottom of the lattice, $y = 0$.

5. Each lattice point that has been visited by the flowing water has its height reduced by a constant amount D . This process represents erosion.
6. Any site at which the height difference Δh with a neighbor exceeds a critical amount M is reduced in height by an amount $\Delta h/S$, where S is another parameter in the model. This process represents mud slides.
7. Steps (2)–(6) are repeated until you wish to analyze the resulting network. The river network is defined as follows. Every site in the lattice receives one unit of precipitation. Then water flows from a site to the nearest neighbor with the smallest height. Then the water flows to the neighbor of this new site with the smallest height. This flow continues until the flow reaches the bottom of the lattice. This process is repeated for each site, and the number of times that a site receives water is recorded. The river network is defined as the set of all sites that received at least R units of water, where R is another parameter of the model.

The adjustable parameters of the model can be related to measurable quantities, and the different steps of the algorithm correspond to real dynamical processes. Hence, we can gain insight into how the results change as we vary the parameters. It would be interesting to program the above model and see what happens. You also would probably understand the algorithm better by converting the algorithm into a working program.

19.4 What are Computers Doing to Physics?

There is probably no need to convince you that computers are changing the way we think about the physical world. The question, “How can I formulate this problem for a computer?” has led to new insights into old problems and is allowing us to consider new ones. The problems of galaxy formation and the evolution of river networks are just two of the many examples from the current research literature.

What will be the effect of computers in physics education? The most common use of computers has been to assist students to understand topics that have been in the curriculum for many years. So far the computer has not qualitatively changed the way we learn nor the topics we study. Will computer simulation and numerical analysis make analytical methods less important? Has this happened already? Should calculus retain its traditional importance in the curriculum? Do we understand a natural phenomenon when we are able to construct a computer model that allows us to make predictions which agree with experiment? Is it necessary to obtain at least some analytical results? What do you think should be the role of computers in education? Now that you have reached the end of this text, we expect that you already have started asking your own questions.

Computers and the visual images produced by computer models can be very seductive. However, we need to remember that the goal of science is to understand Nature. Theory and experiment have been the traditional routes to this end, and computation is now a third and complementary route. Although we have stressed the importance of computation in this text, it is important also to stress its complementary role. We must not let the rapid advances of computer technology and the easy availability of information overshadow our ultimate goal of gaining more knowledge and a deeper understanding of natural phenomena.

Choosing a model depends not just on how realistic you want your simulation to be, but also what kinds of quantities you wish to understand. Cieplak et al. considered two river network models to understand their scaling properties and compare with observational data. One model is based on the invasion percolation model. A lattice is set up with each bond uniformly assigned a random number. The bonds are occupied starting from the bond with the smallest random number, until all bonds have a path to the bottom row of the lattice which models the outlet of the river network where all the water eventually flows. Loops are eliminated as the network is created and the properties of the resulting network are measured. A second model they use follows the Eden model. Here all the bonds have equal probability of being occupied as long as no loops are created. The bonds are occupied at random until all bonds have a path to the outlet. Despite the very different mechanisms in the two models, their properties are similar.

References and Suggestions for Further Reading

It would be impossible to list even a small subset of references to areas of physics and related disciplines that we have not discussed. Also the development of algorithms and applications in areas we have discussed is evolving rapidly. Many references to other applications and current developments can be found in archival journals. Another place to look for recent developments is the various electronic archives available on the Internet. We encourage readers of this text and others who are using computer simulations in related contexts to browse our home page (see the preface), where we hope that many developments will be listed and discussed.

Several references relevant to this chapter are listed in the following.

- A. D. Abrahams, "Channel networks: a geomorphological perspective," *Water Resour. Res.* **20**, 161 (1984). Abrahams criticizes the random walk models of river networks.
- R. M. Anderson, "Population ecology of infectious disease agents," in *Theoretical Ecology: Principles and Applications*, R. M. May, editor, Blackwell (1981).
- Eric Bonabeau and Laurent Dagorn, "Possible universality in the size distribution of fish schools," *Phys. Rev. E* **51**, R5220 (1995). The authors apply a model originally developed for river networks to the size distribution of schools of fish.
- Silvia Burés, David P. Landau, Alan M. Ferrenberg, and Franklin A. Pokorny, "Monte Carlo computer simulation in horticulture: a model for container media characterization," *HortScience*, **28**, 1074 (1993).
- Marek Cieplak, Achille Giacometti, Amos Maritan, Andrea Rinaldo, Ignacio Rodriguez-Iturbe, and Jayanth R. Banavar, "Models of Fractal River Basins," *J. Stat. Phys.* **91**, 1 (1998).
- Michael Creutz, Laurence Jacobs, and Claudio Rebbi, "Monte Carlo computations in lattice gauge theory," *Phys. Repts.* **95**, 201 (1983). See also Michael Creutz, *Quarks, Gluons and Lattices*, Cambridge University Press (1983).
- A. K. Dewdney, "Computer Recreations," *Sci. Am.* **251**(#12), 14 (1984). A discussion of the Wa-Tor model.

Zvonko Fazarinc, “A viewpoint on calculus,” Hewlett-Packard Journal **38**, 38 (March, 1987).

Richard P. Feynman, Robert B. Leighton, and Matthew Sands, *Lectures on Physics, Vol. II*, Addison-Wesley (1964). Our chapter title has been adapted from the subtitle of Section 12.1, “The same equations have the same solutions.”

Robert L. Leheny, “A simple model for river network evolution,” Phys. Rev. **52**, 5610 (1995) and references therein.

Lawrence S. Schulman and Philip E. Seiden, “Percolation and Galaxies,” Science **233**, 425 (1986). A discussion of percolitis is also given. The authors note that for finite N the disease always dies out eventually, but the expected time for the eradication of the disease grows exponentially with N . Also see the review paper Philip E. Seiden and Lawrence S. Schulman, Adv. Phys. **39**, 1 (1990).

Dietrich Stauffer, editor, *Annual Reviews of Computational Physics I and II*, World Scientific (1995).