COMPUTER RECREATIONS

Author(s): A. K. Dewdney

Source: *Scientific American*, Vol. 259, No. 2 (AUGUST 1988), pp. 104-107
Published by: Scientific American, a division of Nature America, Inc.
Stable URL: https://www.jstor.org/stable/10.2307/24989205

COMPUTER RECREATIONS

The hodgepodge machine makes waves

by A. K. Dewdney

Cellular automatons, computer models based on arrays of multivalued cells, have spread like a wave through physics, mathematics and other sciences. Now a new cellular automaton has literally been making waves of its own. Called the hodgepodge machine by its designers, it imitates chemical reactions with a precision rarely seen in other models.

The reactions the hodgepodge machine simulates take place in excitable chemical mediums: two or more compounds that can dissociate and recombine in the presence of a catalyst. If the chemical states of the reactants have different colors, wavelike structures can be seen that propagate along simple or intricate frontiers in endless pursuit of an elusive equilibrium.

Does the automaton itself serve as an adequate physical explanation for the waves observed in actual reactions? This question now occupies the hodgepodge machine's creators, Martin Gerhardt and Heike Schuster of the University of Bielefeld in West Germany, along with an increasing number of colleagues at other universities.

A cellular automaton can be thought of as an infinite grid of square cells that advance through time in step with discrete ticks of an imaginary clock. At any given tick each cell is in one of a finite number of states. The state of a cell at tick $t+1$ depends in a fairly simple way on the states of the cells in its immediate neighborhood at the previous tick, t. The dependence is expressed in a set of rules that apply equally to all the cells in the grid. By applying the rules each time the clock ticks, an arbitrary initial configuration of states among the cells can be made to change and thus evolve with time. In some cases extraordinary patterns develop, prompting observers to believe that given the right initial configuration a cellular automaton could produce something capable of organizing itself, growing and reproducing—in short, something "living."

The cellular automaton best known to readers is probably the famous game of Life invented in the 1960's by the mathematician John Horton Conway of the University of Cambridge. In Life each cell has only two possible states: alive and dead. The rules of Life are very simple. If a cell is dead at time t, it will come to life at time $t+1$ if exactly three of its neighbors are alive at time t. If a cell is alive at time t, it will die at time $t+1$ if fewer than two or more than three of its neighbors are alive at time t. These two rules are sufficient for the Life cellular automaton to display an amazing variety of behavior that depends entirely on the configuration of dead and alive cells with which one starts [see "Computer Recreations," SCIENTIFIC AMERICAN, May, 1985, and February, 1987].

The hodgepodge machine is not one cellular automaton but many. One chooses a particular version by specifying a number of parameters such as the number of states. If there are $n+1$ states, each possible state of a cell can be represented by a number between 0 and n. Gerhardt and Schuster extend Conway's metaphor to describe the states of the cells in their machine. A cell in state 0 is said to be "healthy" and a cell in state n is said to be "ill." All states in between exhibit a degree of "infection" corresponding to their state number; the closer a cell's state number gets to n, the more infected the cell becomes. The hodgepodge machine selectively applies one of three rules to each cell, depending on whether it is healthy, ill or infected.

If the cell is healthy (that is to say, in the 0 state), at the next tick of the clock it will have a new state that depends on the number of infected cells, A, and the number of ill cells, B, currently in its neighborhood and on two parameters labeled $k1$ and $k2$. To be specific, the state of the cell at time $t+1$ is given by the following formula:

$$[A/k1] + [B/k2].$$

A pair of square brackets designates a rounding-down process applied to the fraction it contains. If, for example, $A/k1$ happens to equal 2.725, the square brackets reduce that number to 2. If the formula happens to yield a 0, the cell will of course remain healthy—at least for the time being.

If the cell is infected, its condition generally worsens with time. Its state at time $t+1$ is the sum of two numbers: the degree of infection in the cell's neighborhood at time t and an unvarying quantity, g, that governs how quickly infection tends to spread among the cells. The degree of infection is calculated by dividing S, the sum of the state numbers of the cell and of its neighbors, by A, the number of infected neighbors. A cell in an infected state at time t therefore takes on at time $t+1$ a state given by the formula

$$[S/A] + g.$$

The infected cell cannot get "sicker" than n, however. If it happens that the number given by the formula exceeds n, then n is taken to be the new state of the cell.

Finally, if the cell is ill (in state n) at time t, it miraculously becomes healthy (takes on a state of 0) at $t+1$.

In addition to those three rules a definition of what constitutes a cell's "neighborhood" is necessary. Two types of neighborhood have historically been used in cellular automatons: the von Neumann neighborhood and the Moore neighborhood. The von Neumann neighborhood of a particular cell consists of the four cells that share the cell's edges. The Moore neighborhood of a particular cell includes the cells in the von Neumann neighborhood and also the four cells that just touch the cell's corners—a total of eight cells. Given the three rules and the definition of a cell's neighborhood, the Gerhardt-Schuster cellular automaton is completely defined by specifying the values of four parameters: n, the number of states minus 1; $k1$ and $k2$, the "weighting" parameters for healthy cells, and g, the speed of infection.

A sample experiment done by Gerhardt and Schuster on a 20-by-20 grid using von Neumann neighborhoods reveals the typical behavior of hodgepodge machines. (Cells at the edge of the grid abide by the same rules that prevail elsewhere in the cellular automaton; they just have fewer cells in their neighborhood.) The parameters n, $k1$ and $k2$ were fixed respectively to the values of 100, 2 and 3. Four types of behavior emerged at different values of the parameter g. In a typical trial run Gerhardt and Schuster gave the 400 cells in the 20-by-20 grid a random initial configuration of states, specified a value of g and let the hodgepodge machine loose for 10,000 computational cycles. Because one-dimensional data are easier to analyze than two-dimensional images, Gerhardt and Schuster recorded only the number of infected cells at each cycle in order to present their results in graphs like those on the next page.

Not much happened to this hodgepodge machine at low g values. Apart from a few initial fluctuations, activity among the cells tended to die out; the cells became boringly and everlastingly healthy. But as g was increased, strange things began to happen. To begin with, most of the cells became infected and remained so, although there were irregular and random appearances of healthy cells. Gerhardt and Schuster labeled this type of behavior Type 1.

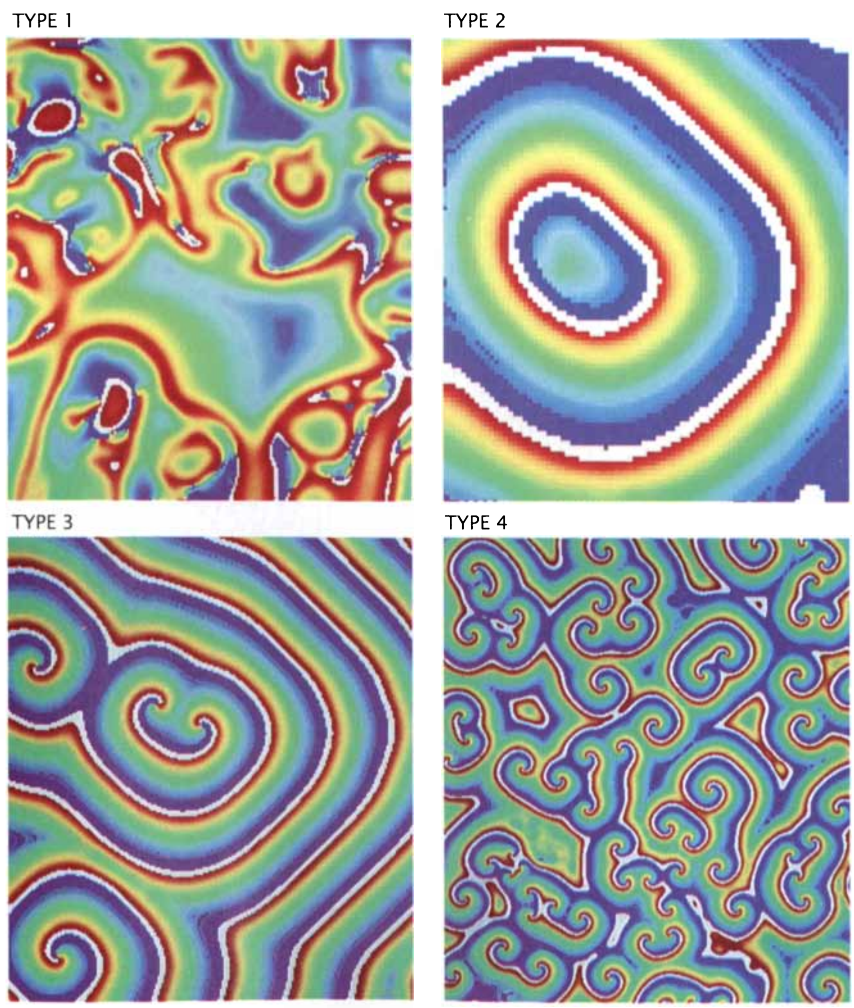
The next type of behavior they observed was labeled Type 2. It featured a generally regular series of infection "plateaus" roughly 30 cycles long, punctuated by the appearance of large numbers of healthy cells. (Sometimes nearly all 400 cells became healthy only to experience a new wave of infection.) As g was increased still further, Type 3 behavior appeared. It was heralded by the onset of a very regular alternation between saturation and virtual disappearance of infected cells every 20 cycles or so. Finally, Type 4 behavior emerged: within a few cycles of start-up the number of infected cells would fluctuate with some regularity about a saturation value of approximately 75 percent.

The four types of behavior appeared in order as g was progressively increased, but with some overlap: runs with transition values of g sometimes resulted in one type of behavior and sometimes in another type. In certain cases Gerhardt and Schuster even witnessed transitions between behaviors in a single run.

The four behaviors represent the appearance of specific types of wave patterns that are shown in the illustration below. In those color images the grid sizes vary from 100-by-100 cells to 500-by-500 cells. Waves associated with Type 1 behavior traveled only a short distance before dying out. Type 2 waves traveled outward in circular bands that varied greatly in width. Type 3 waves displayed the same circular shape but were more regular, in keeping with the regular ups and downs of infected cells displayed in its graph. Finally, Type 4 waves followed a spiral pattern that spread out from the center of the grid. As always, readers with computers are urged to repeat the experiment in some form. Waves of thought are sure to accompany the waves on one's screen.

Some of the wave patterns generated by the hodgepodge machine are similar to those displayed by a variety of chemical systems; certain ones in particular are dead ringers for the chemical waves found in the well-known Belousov-Zhabotinsky reaction. Compare, for example, the complex pattern of curlicues in the computer-generated image with the photograph of the Belousov-Zhabotinsky reaction in the illustration on page 107.

To what do we owe this similarity? Gerhardt and Schuster were not exactly surprised by it; they had deliberately designed the hodgepodge machine to mimic the features of a particular kind of "heterogeneous catalytic reaction" in which carbon monoxide and oxygen combine to form carbon dioxide while adsorbed at the surface of thousands of tiny palladium crystallites dispersed throughout a porous medium. Heat given off as the oxidation reaction proceeds changes the state of the catalyst. An abrupt phase transition by the crystallite liberates the carbon monoxide adsorbed at its



The hodgepodge machine produces distinctive wave patterns

surface; the catalyst then cools and the reaction begins anew.

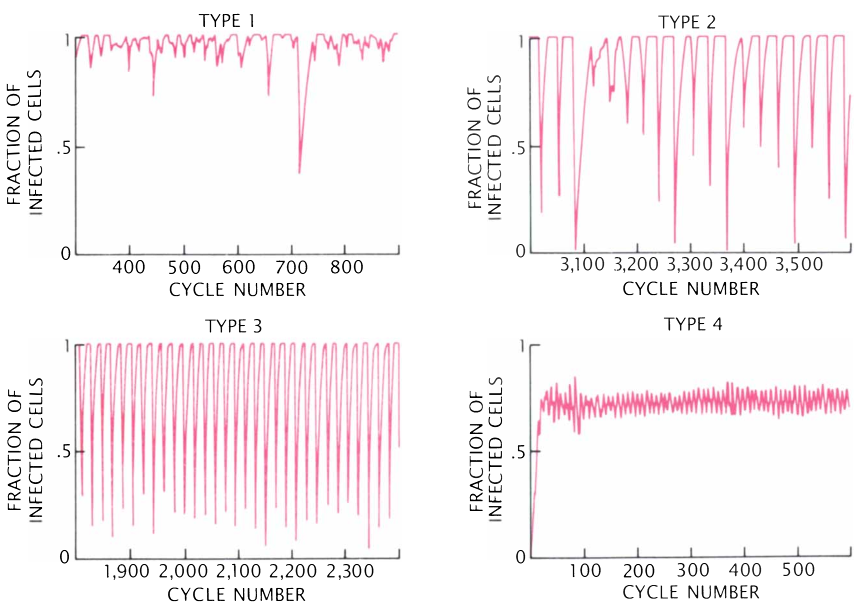
The hodgepodge machine proved capable of mimicking not only this reaction but also the Belousov-Zhabotinsky reaction quite well. In the Belousov-Zhabotinsky reaction malonic acid is oxidized by potassium bromate in the presence of a catalyst such as cerium or iron. The grid cells of the hodgepodge machine in essence represent the catalyst particles, and the infection metaphor expresses the gradual saturation of the particles' surfaces.

But the analogy is not quite so simple; there are some subtleties here. For one thing, in the hodgepodge machine adjacent cells interact by exchanging infection, so to speak. How do the catalyst particles exchange reactivity? Gerhardt and Schuster reasoned that, at least in the case of the carbon monoxide oxidation, the participating catalyst units influence their neighbors by means of two basic mechanisms. A given unit could be made more reactive by the transfer of heat from a more active neighboring unit or by the diffusion of carbon monoxide from a less active neighbor.

The interaction between neighboring cells in the hodgepodge machine makes it possible for them to synchronize their activities. After a period of initial random disorganization (the hodgepodge phase), the patterns that appear reflect this synchronization. The same is presumably true of the actual chemical reactions as well. Does the hodgepodge machine thus explain the appearance of waves of excitation in the reactions it simulates?

There will be those who are ready to exclaim "Of course!" and to point to the pictures as evidence. But then, there are people who see a cellular automaton in everything. In April *The Atlantic* carried an article about the cosmic ramblings of Edward Fredkin. A computer businessman and sometime academic, Fredkin supposes our universe to be composed of cells that tick from state to state like a vast cellular automaton. To be kind, the evidence for such an arrangement is not overwhelming. The hodgepodge machine is doubtless significant, but the attitude of its discoverers is more so. In spite of the fact that the hodgepodge machine simulates the Belousov-Zhabotinsky reaction remarkably well, Gerhardt and Schuster do not claim that chemistry is cellular. Instead they see their automaton as an approximation tool, the discrete version of a partial differential equation.

Originally inspired by the work of chemists Nils Jaeger and Peter Plath of the University of Bremen, Gerhardt and Schuster along with their mentor at Bielefeld, Andreas W. M. Dress, have enlisted the help of two chemists in studying the hodgepodge machine: S. C. Müller of the Max Planck Institute for Nutritional Physiology in Dortmund and John J. Tyson of the Virginia Polytechnic Institute and State University. The creators of the machine want to show that an array of chemical oscillators that interact locally according to certain simple rules will inevitably generate waves. Presumably there are only a small number of possible wave patterns, although they become far more complicated in three dimensions, according to Tyson. Because three-dimensional wavefronts are much harder to see in laboratory glassware, computer simulations may tell chemists what to look for. In science the trick is to use models well, not to be used by them.



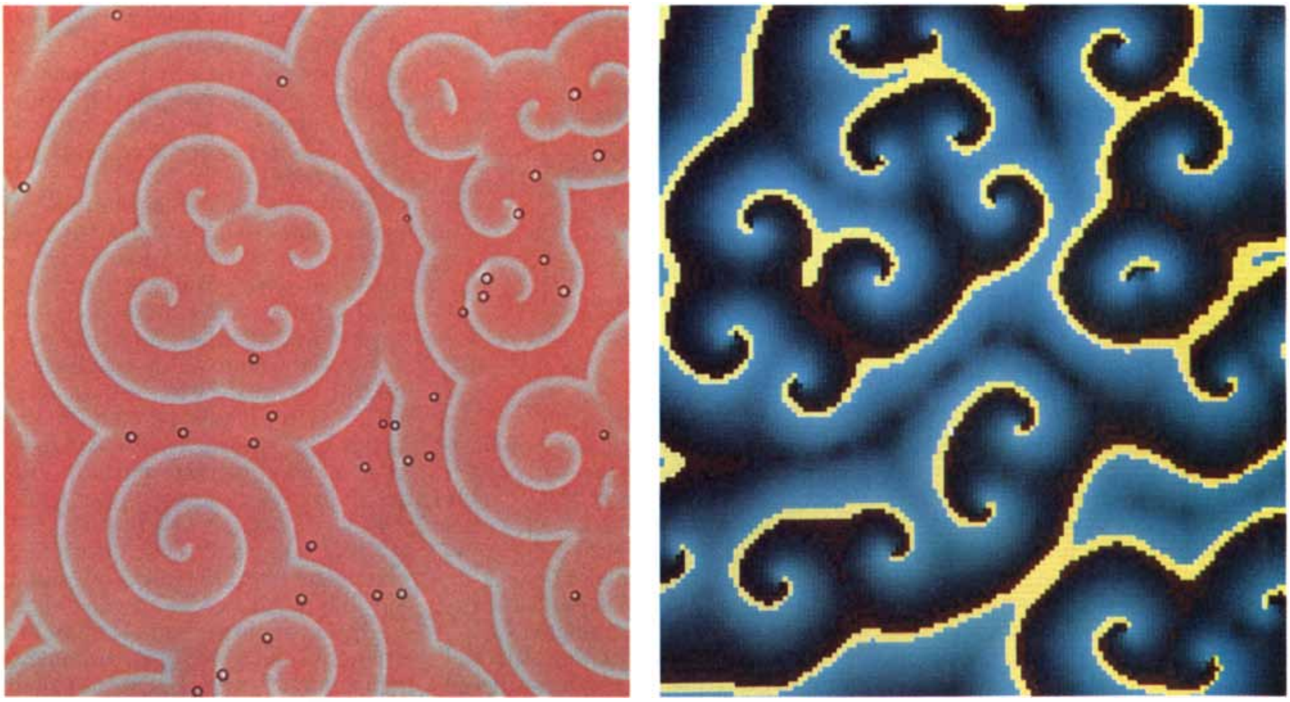
The four behaviors of the hodgepodge machine

Readers who would like to build their own hodgepodge machine have already received ample hints on how to proceed. One must declare an array of appropriate size and incorporate it into a grand loop that updates the array according to the three rules and then displays it for the edification of local hodgepodgers. Each element of the array must contain the state number for a particular cell. In computing the updated array, however, it is necessary to store the results temporarily in another array until the computation is complete. Then a simple double loop allows wholesale replacement of the original array by the updated one.

The updating is also carried out by a double loop. Two index variables, say i and j, count off the cells of the grid. For each cell given by the coordinates (i,j), the program (can we call it anything but HODGEPODGE?) decides by means of a pair of "if" statements whether the cell is healthy or infected. If it is healthy, the first formula is evaluated. If it is infected, the second formula is evaluated. In either case the states of the cells in its neighborhood must be checked. If the cell is neither healthy nor infected, it is obviously ill and will recover at the next cycle.

For reasons of space I am limited to this brief recipe. Readers who would like a more complete algorithmic description of the hodgepodge machine should write to me in care of this magazine. Please include a check or money order for $2 to cover postage (worldwide), copying and other costs.

The Apraphulian excursion of April fooled few people. Those who nonetheless entered into the spirit of the account were challenged by the reconstruction of the Apraphulian analog multiplying machine: a device that multiplies two numbers entirely by means of ropes and pulleys. Some entered into the spirit of the enterprise so fully that they asserted they had firsthand knowledge of the ancient Apraphulian culture. The champion letter in this vein was sent in by Clive J. Grant of Chichester, N.H. A long document describes Grant's correspondence with a



Wave phenomena in a Belousov-Zhabotinsky chemical reaction (left) _and their hodgepodge counterparts_ (right)

mysterious Dr. Ebur Grebdlog, renowned scholar of Apraphulian lore:

"After reading your 'Computer Recreations' column on the state of Apraphulian mathematics, I contacted Dr. Grebdlog to ask him if his work had ever covered. . .an Apraphulian Analog Multiplier.... Indeed, he replied, he had investigated that matter, and he sent along a copy of his work."

The "work" was beautifully written in Grebdlog's spidery hand, accompanied by technical drawings of pulleys and cams connected by bridges. Grebdlog notes that the drawings "appear to have guided the Apraphulians in constructing a device truly remarkable for the unstinting technological effort applied to its development but more remarkable for its total lack of utility."

The multiplier most often suggested by readers made use of a rod one end of which is attached to a fixed hinge. An input rope tied partway along the rod pulls it forward so that an output rope tied to the free end is also pulled in the same direction. Since the rod is in essence a lever with the fulcrum at its hinged end, the output rope moves a greater distance than the input rope. The problem with this design arises from a loss of proportionality: as the input rope is pulled farther, the rod follows a circular arc and the amplifying effect on the output rope eventually fades. Variations on this theme sometimes corrected for the rod's circular motion by means of guides or fancy systems of parallel jointed rods. All of this struck me as too complicated. Perhaps I should have explicitly forbidden the use of rods.

Robert Norton of Madison, Wis., used spiral pulleys to compute logarithms and antilogarithms. The input ropes *A* and *B* are unwound from two drums (which are not against the rules, since they are just wide pulleys). Each drum is attached to a spiral drum that winds up an output rope. The two outputs are then added in the way outlined at the end of the April column. The antilog of the sum is computed by winding the addition rope onto a spiral drum that is connected to a straight drum on which the final output rope is wound. A similar machine was "discovered" by Robert A. Eddius of New York City. The Apraphulians, he contends, used the shells of certain mollusks whose spiral shape enabled them to compute logarithms exactly! On the other hand, David A. Fox of Lima, Ohio, writes us that a similar culture inhabited a small island off the Marshall group known as Hardly Atoll. Here were found not only the same log-antilog devices but also a contraption rather like a yo-yo that was capable of squaring numbers. Readers might want to ponder whether Fox's assertion is possible.

Caxton C. Foster of East Orleans, Mass., is of the opinion that the Apraphulian civilization was destroyed by logical gain: the problem encountered by a computer in which the "1" output of each gate is not quite 1. To prevent such inaccuracies from creeping into the sacred computations, the high priests stationed an Apraphulian at each gate to pull a little harder on any output ropes lacking the necessary tautness. Thus absorbed, the people were unable to procure food and eventually starved to death.

The final word belongs to modern-day computer architect Michael Pagan of Mount Laurel, N.J. Concerned about the cultural gap between the analog branch and the digital branch of Apraphulian society, Pagan developed a marvelous analog-to-digital converter. A single rope carrying the analog signal enters the device and a number of ropes bearing the digital equivalent of the input number leave it. Such a machine may have been introduced on Apraphul, but the priests would certainly have banned the pagan device.

FURTHER READING

THE ARMCHAIR UNIVERSE. A. K. Dewdney. W. H. Freeman and Company, 1988.

DID THE UNIVERSE JUST HAPPEN? Robert Wright in *The Atlantic*, Vol. 261, No. 4, pages 29–44; April, 1988.