Chapter 12

Game of Life

Conway's Game of Life makes use of sparse matrices.

The "Game of Life" was invented by John Horton Conway, a British-born mathematician who is now a professor at Princeton. The game made its public debut in the October 1970 issue of *Scientific American*, in the "*Mathematical Games*" column written by Martin Gardner. At the time, Gardner wrote

> This month we consider Conway's latest brainchild, a fantastic solitaire pastime he calls "life". Because of its analogies with the rise, fall and alternations of a society of living organisms, it belongs to a growing class of what are called "simulation games" – games that resemble real-life processes. To play life you must have a fairly large checkerboard and a plentiful supply of flat counters of two colors.

Of course, today we can run the simulations on our computers.

The *universe* is an infinite, two-dimensional rectangular grid. The *population* is a collection of grid cells that are marked as *alive*. The population evolves at discrete time steps known as *generations*. At each step, the fate of each cell is determined by the vitality of its eight nearest neighbors and this rule:

- A live cell with two live neighbors, or any cell with three live neighbors, is alive at the next step.

The fascination of Conway's Game of Life is that this deceptively simple rule leads to an incredible variety of patterns, puzzles, and unsolved mathematical problems – just like real life.

If the initial population consists of only one or two live cells, it expires in one step. If the initial population consists of three live cells then, because of rotational

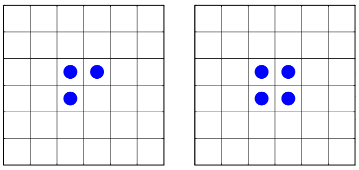

Figure 12.1. *A pre-block and a block.*

and reflexive symmetries, there are only two different possibilities – the population is either L-shaped or I-shaped. The left half of figure 12.1 shows three live cells in an L-shape. All three cells have two live neighbors, so they survive. The dead cell that they all touch has three live neighbors, so it springs to life. None of the other dead cells have enough live neighbors to come to life. So the result, after one step, is the population shown in the right half of figure 12.1. This four-cell population, known as the *block*, is stationary. Each of the live cells has three live neighbors and so lives on. None of the other cells can come to life.

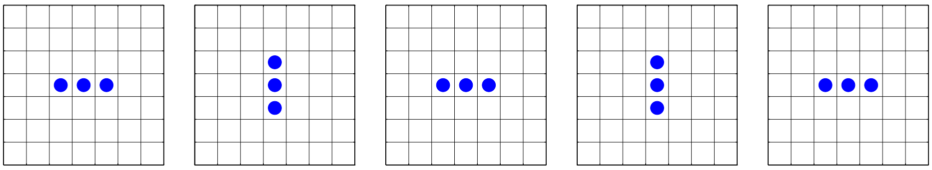

Figure 12.2. *A blinker blinking.*

The other three-cell initial population is I-shaped. The two possible orientations are shown in first two steps of figure 12.2. At each step, two end cells die, the middle cell stays alive, and two new cells are born to give the orientation shown in the next step. If nothing disturbs it, this *blinker* keeps blinking forever. It repeats itself in two steps; this is known as its *period*.

One possible four-cell initial population is the block. Discovering the fate of the other four-cell initial populations is left to an exercise.

The beginning of the evolution of the most important five-cell initial population, known as the *glider*, is shown in figure 12.3. At each step two cells die and two new ones are born. After four steps the original population reappears, but it has moved diagonally down and across the grid. It continues to move in this direction forever, eventually disappearing out of our field of view, but continuing to exist in the infinite universe.

The fascination of the Game of Life cannot be captured in these static figures. Computer graphics lets you watch the dynamic development. We will show just more one static snapshot of the evolution of an important larger population. Figure 12.4 is the *glider gun* developed by Bill Gosper at MIT in 1970. The portion of the population between the two static blocks oscillates back and forth. Every 30 steps, a glider emerges. The result is an infinite stream of gliders that fly out of the field of view.

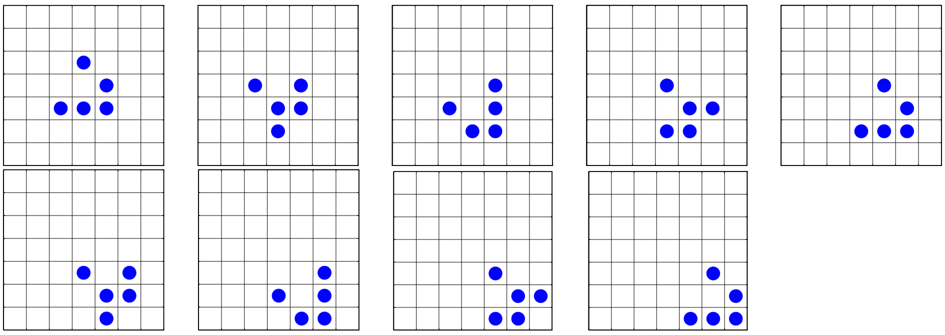

Figure 12.3. *A glider gliding.*

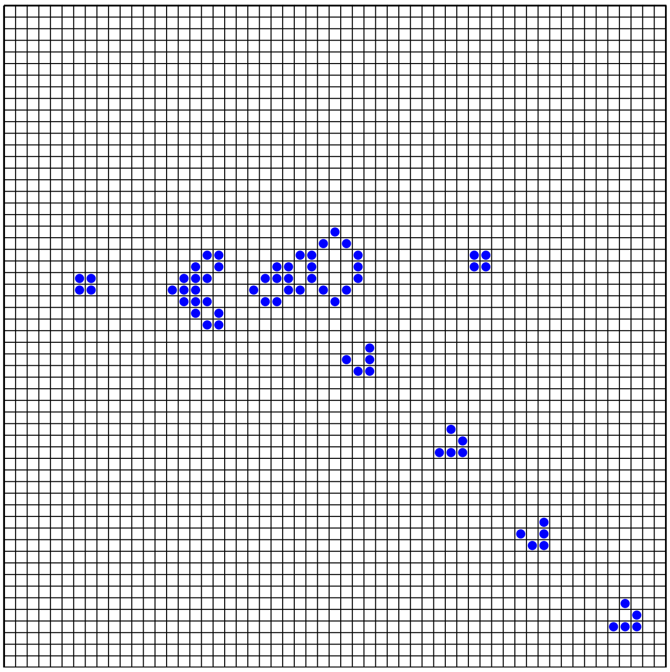

Figure 12.4. *Gosper's glider gun.*

Matlab is a convenient environment for implementing the Game of Life. The universe is a matrix. The population is the set of nonzero elements in the matrix. The universe is infinite, but the population is finite and usually fairly small. So we can store the population in a finite matrix, most of whose elements are zero, and increase the size of the matrix if necessary when the population expands. This is the ideal setup for a *sparse* matrix. Conventional storage of an n-by-n matrix requires n^2 memory. But sparse storage of a matrix `X` requires just three vectors, one integer and one floating point vector of length `nnz(X)` – the number of nonzero