Mathematica Problems on Recurrence Relations (RR) and Cellular Automata (CA)

Ibrahim Jah, Leonardo Rochinha Pedro

May 2019

Problem 1

You take a loan of S dollars that is to be paid back in T periods of time. If r is the interest rate per period of the loan, what constant payment P do you have to make at the end of each period. This is a boundary value problem. What are the boundary values? Solve it on the computer. Say r = 0.06 and S = 15000. Try different P and see for which T you have paid back. Plot with command Plot how the debt changes with time. Find a formula for P(r, s, T).

Solution

If a constant payment of 900 is made the loan wont be paid off as shown below.

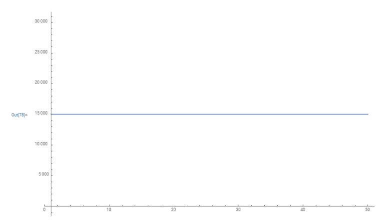

Figure 1: Constant payment of 900

With a constant payment of 950 we are able to eventually pay back the loan.

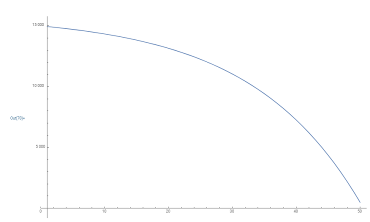

Figure 2: Constant payment of 950

Problem 2

Check if you, like in the logistic map, can find a stable 2-cycle for the map $g(x) = x\ e^{a(1-x)}$. The command for exponential function is Exp. Starting value is now any positive real number but you can take something close to 1. Interesting a values are between one and five. You have to change function in the program. Solve will protest somewhat when you run but you can trust the output. (You can take away the Solve-part). You also have to change the plot ranges.

Solution

As in the logistic map we can find a stable 2-cycle for the map $g(x) = x\ e^{a(1-x)}$, as we started by taking a look at the window with range of $1 \leq$ a ≤ 4 where we find a bifurcation Diagram. This bifurcation Diagram had coordinates of a=2.52 and start = 1.73.

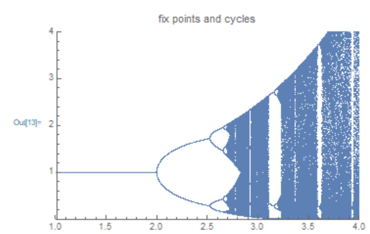

Figure 3: Bifurcation Diagram

With this data we were able to get the orbit for these coordinates where we see the population almost goes extinct but instead manages to stay stable.The orbit graph below shows us the 500 iteration of the orbit for a=2.52 at a start=1.73.

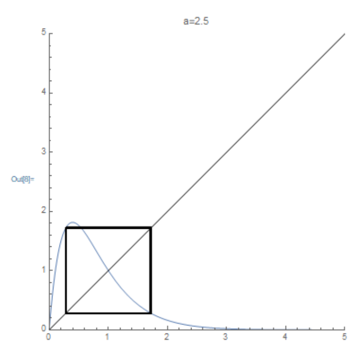

Figure 4: Orbit Diagram

Problem 3

Start with one black cell ■.Create a triangle with alternating black and white cells. So next row is ■□■ and then ■□■□■ and so on. You have to explain how you get the rule number.

Solution

The pattern is made with the use of rule 50. In Cellular Automata (CA) to find out the rule of a pattern you first write out 8 different bit patterns which are possible. Cells in CA can represent being alive or dead, in binary terms it is either a 1(black) or 0(white). The 8 patterns written down in binary form are 111, 110, 101, 100, 011, 010, 001 and 000.

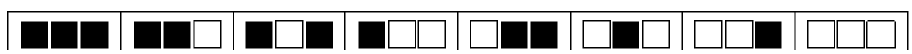

Figure 5: generating cells

The next step in finding our bit pattern is filling out the bottom cells of the 8 possible patterns. The bottom cells can be found out if we look at the the CA pattern and if there's a pattern that exists the bottom cell becomes a 1 otherwise it is a 0. With all the tools needed the next step is to find out the

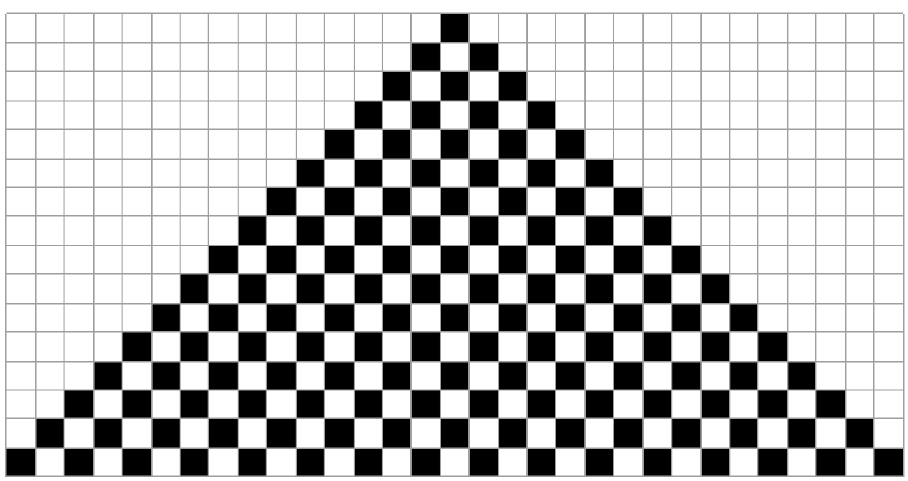

Figure 6: cellular automata

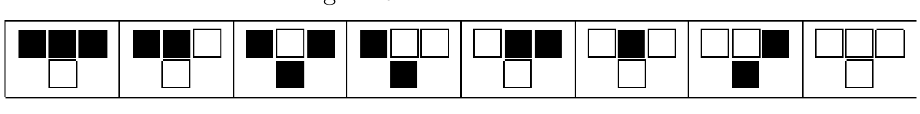

Figure 7: 8 cell generations

rule for the pattern which is the sum of the decimal values of from the binary patterns used so 32+16+2 = 50.

Problem 4

Investigate the rules B1/S and B2/S for various seeds. Also random seeds (use RandomInteger command). Determine the rule numbers. No survival for these two rules.

Solution

B1/S and B2/S are both explosive rules in which every cell dies in every generation. Rule B2/S is a life-like cellular automaton called "Seeds" in which only dead cells with exactly two live neighbours will turn into live cells on the next generation. Even though all the living cells die in every generation (turning every pattern into a phoenix), most patterns are still exploding quadratically. Since it is an exploding rule where every cell dies in every generation. It has many simple orthogonal spaceships, though it is in general difficult to create patterns that don't explode. When the rule B1/S is applied on the cube see (2x2) we end up with a simple expansion pattern where the corner cells live, and the original cells die. On the next step all the surrounding cells turn into living cells. During the next step the outer cells move out just like the first step then all other cells end up dying. This pattern is repeated.

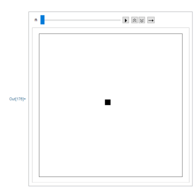

Figure 8: Step 1 of B1/S

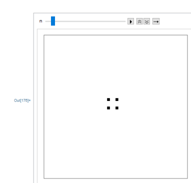

Figure 9: Step 2 of B1/S

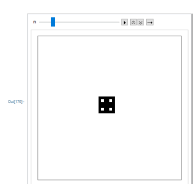

Figure 10: Step 3 of B1/S

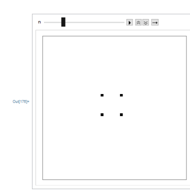

Figure 11: Step 4 of B1/S

When the rule B2/S is applied on the cube seed the all surrounding cells become living cells except for the corner ones, the original cells then die. The same thing happens as the previous step where all surrounding cells come alive apart from the corner ones and the original ones die. The same thing keeps happening for another step but in the step afterwards the inner cells start coming to live. B2/S causes a much more complex pattern.

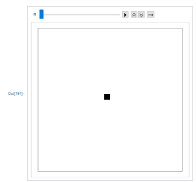

Figure 12: Step 1 of B2/S, Cube

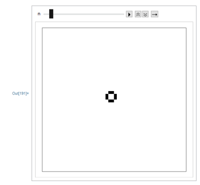

Figure 13: Step 2 of B1/S, Cube

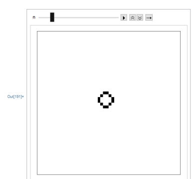

Figure 14: Step 3 of B2/S, Cube

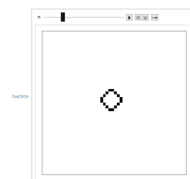

Figure 15: Step 4 of B2/S, Cube

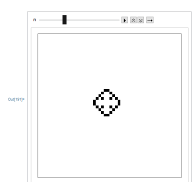

Figure 16: Step 5 of B2/S, Cube

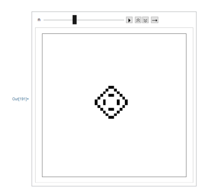

Figure 17: Step 6 of B2/S, Cube

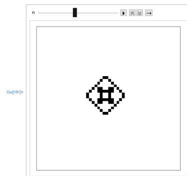

Figure 18: Step 7 of B2/S, Cube

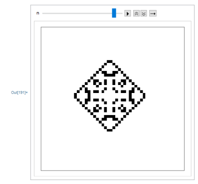

Figure 19: Step 8 of B2/S, Cube

When we applied B1/S to the Beehive seed we ended up with a complex expansion which led to a square shaped expansion with complex patterns inside it.

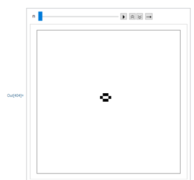

Figure 20: Step 1 of B1/S, Beehive

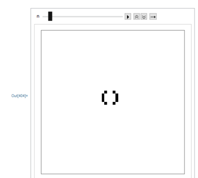

Figure 21: Step 2 of B1/S, Beehive

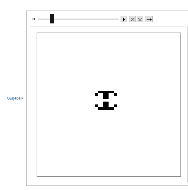

Figure 22: Step 3 of B1/S, Beehive

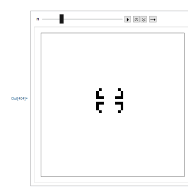

Figure 23: Step 4 of B1/S, Beehive

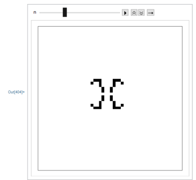

Figure 24: Step 5 of B1/S, Bee-hive

B2/S on the Bee-hive seed caused all surrounding cell to live except for corner cells, the original cells then died. The second step caused the top half of the pattern to move up a cell and the bottom half to move down a cell, then a line formed between them. In the third step the line of cells died and the rest of the pattern expanded in the same direction, this occurrence keeps happening. which led to a square shaped expansion with complex patterns inside it.

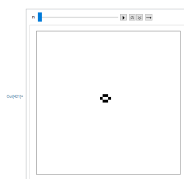

Figure 25: Step 1 of B2/S, Bee-hive

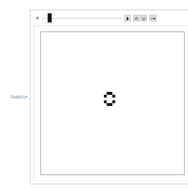

Figure 26: Step 2 of B2/S, Bee-hive

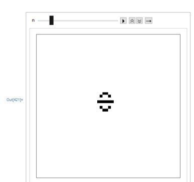

Figure 27: Step 3 of B1/S, Bee-hive

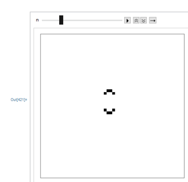

Figure 28: Step 4 of B1/S, Bee-hive

When generating random seeds with the 'RandomInteger' command for both rule B1/S and B2/S we observed a large amount of cells dying after the first step due to overpopulation and then in the following step there was a small growth, where the cells then managed to remain at a stable quantity for every step after the first two.

Figure 29: Step 1 of generated seeds

Figure 30: Step 2 of generated seeds

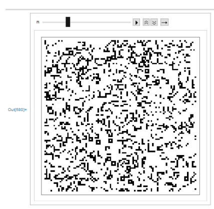

Figure 31: Step 15 of generated seeds

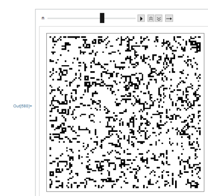

Figure 32: Step 30 of generated seeds