

Figure 7.5.3 Evolution of a Asymmetric Neighborhood Cyclic Cellular Automaton

7.6 Experiment: The Hodgepodge Rule

In this section we look at rule that is of a type used to model the spread of illnesses and which is similar to rules used to describe chemical reactions involving catalysts. While this Hodgepodge rule is traditionally viewed as being based upon cells in one of many states (say 100), we will look at a slight variant that uses fuzzy values for the intermediate states.

The basic idea of the rule is that each cell is either healthy (value 0), ill (value 1) or infected (intermediate between 0 and 1). A healthy cell becomes infected if a threshold of ill or infected neighbors occur. An infected cell, moves toward being ill by a fixed amount from an average of its neighbors. An ill cell becomes healthy on the next generation. We will apply these rules on 3 by 3 neighborhoods, but we will ravel them before processing, so we can view them as length 9 vectors. Below we generate a matrix `X` that contains 3 neighborhoods, one in each row, containing one each of healthy, infected and ill center cells. The function `case` determines state of the center cell in each neighborhood, resulting in 0, 1, or 2 depending whether the cell is healthy, infected or ill.

```
]X=: (t=1)>. (0=t=:?.3 9$5)*(?.3 9$0)
0      1 0 1 0.0583756 0 1 1      1
0      0 0 0      1 0 0 0 0.119496
0 0.392216 0 0      0 0 0 0      0

case=: (~:&0 + =&1)@(4&{ )
case 0{X      infected
1
case 1{X      ill
2
case 2{X      healthy
0
```

The evolution of the rule depends upon four parameters, `a`, `b`, `c` and `N`. Healthy cells become infected if the number of infected or ill neighbors passes a threshold, the level of infection is determined by the number of times the thresholds are exceeded, added together and scaled back by `N` to remain between 0 and 1.

```
ill=: #~ =&1
inf=: #~ ~:&0 *. ~:&1

ill 0{X      ill part of the neighborhood
1 1 1 1 1
```

```

    inf 0{X
0.0583756
    'a b c N'=.2 3 0.17 100
    forh1=. <.@:(*&(%a))@#@ill
    forh2=. <.@:(*&(%b))@#@inf
    forhea =. [: %&N forh1 + forh2
    forhea 0{nei
0

```

infected part of the neighborhood

this healthy cell stays healthy

Next, infected cells tend to increase in level of infection, toward illness, as the average of infected neighbors plus an additional infection amount, *c*, and the total not exceeding one.

```

    avg=:+ / % #
    forinf=: 1 <. avg@inf + c"_
    forinf 2{nei
0.228376

```

The ill cells immediately revert to healthy. We can put all of this into an adverb for creating a local Hodgepodge rule.

```

    lhodge=: 1 : 0
'a b c N'=.m
ill=. #~ =&1
inf=. #~ ~:&0 *. ~:&1
case=. (~:&0 + =&1)@(4&{)
avg=.+ / % #
forh1=. <.@:(*&(%a))@#@ill
forh2=. <.@:(*&(%b))@#@inf
forhea =. [: %&N forh1 + forh2
forinf=. 1 <. avg@inf + c"_
forill=. 0:
forhea`forinf`forill@.case@:, f.
)

```

defined by *automata.ijs*

```

    2 3 0.17 100 lhodge 0{X
0.228376
    2 3 0.17 100 lhodge 1{X
0
    2 3 0.17 100 lhodge 2{X
0

```

Now we can make this into a global rule and use `show_auto` to display the evolution on a random 200 by 200 array of cells.

```

VRAWH=:600 600
b=: (c=1)>.(0=c=:?.200 200$20)*(?.200 200$0)
hodge=: 1 : '3 3&(m lhodge;._3)@ perext2'

```

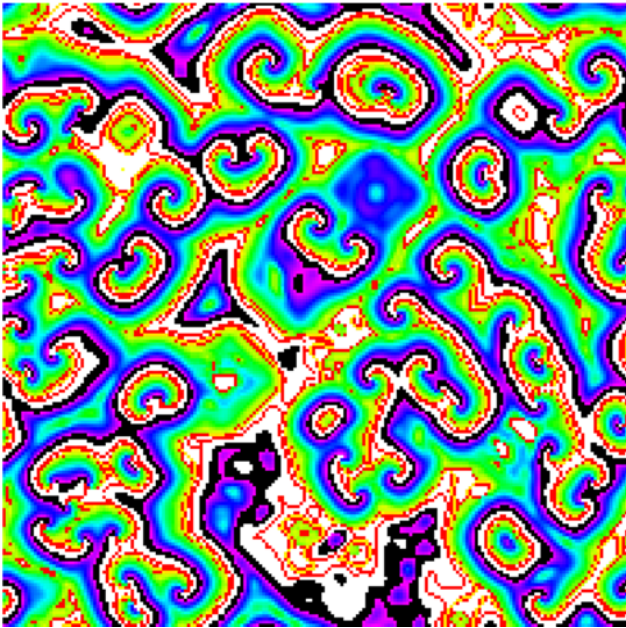


Figure 7.6.1 Hodgepodge Gives Spirals

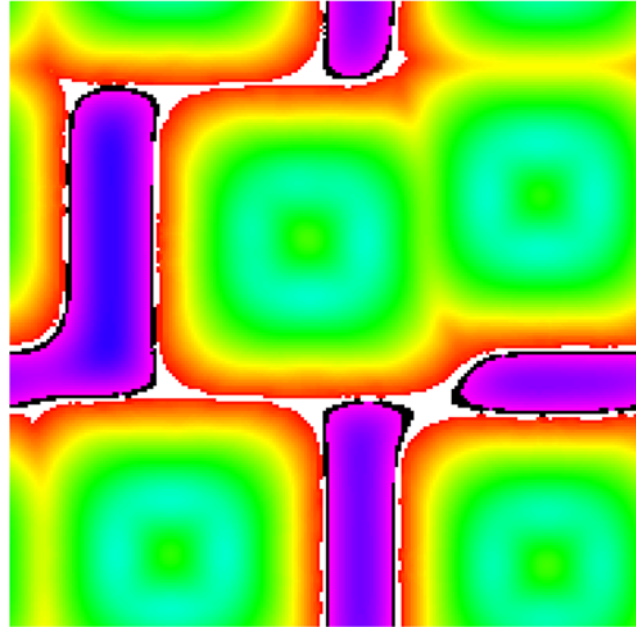


Figure 7.6.2 Hodgepodge Structures

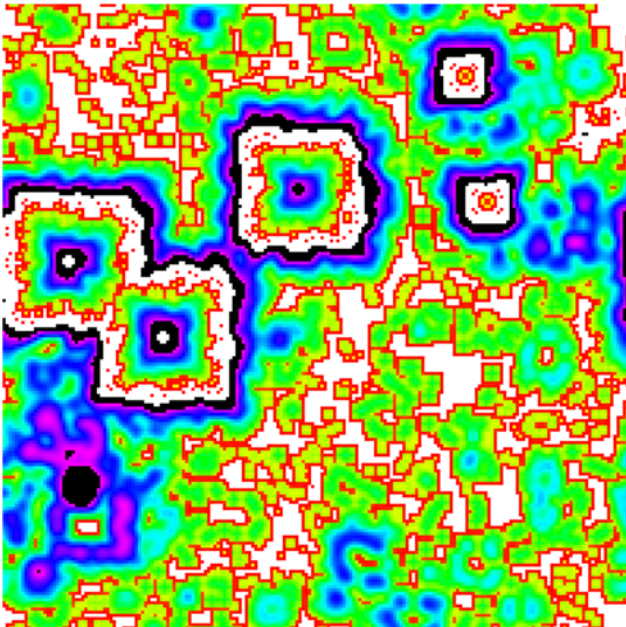


Figure 7.6.3 Hodgepodge Square Waves

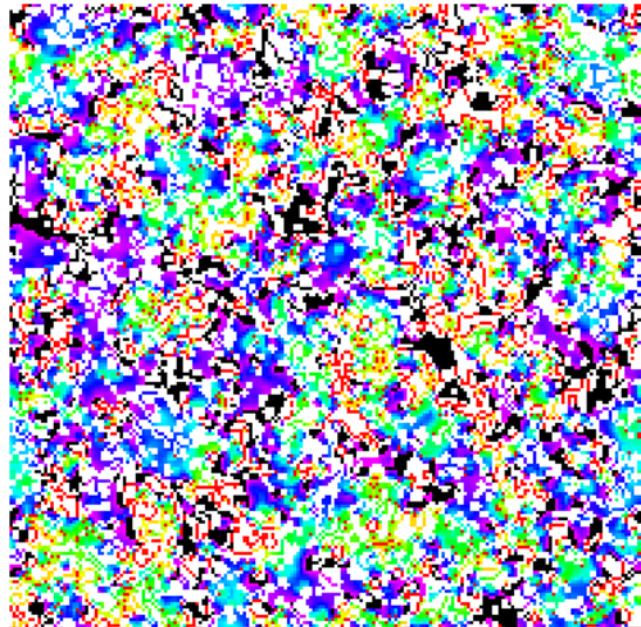


Figure 7.6.4 Hodgepodge is Grainy

```
200 0.1 (2 3 0.17 100 hodge) show_auto b
0.1
```

Watching the evolution shows moving wave-fronts. White corresponds to healthy cell, hues to infections, and black as ill. The last step of that evolution is shown in Figure 7.6.1. Notice that the ends of the wave-fronts exhibit some dramatic spiraling arms. Watching these develop for more than 200 steps can be fun too. Figure 7.6.2 shows a highly structured configuration resulting from parameters 6 1 0.02 100. Figure 7.6.3 shows very rich structured configuration resulting from the parameters 4 1 0.2 100. Watching that hodgepodge machine develop for more than 200 iterations is worthwhile too. Figure 7.6.4 shows very fine patterns resulting from parameters 2 6 0.16 100. The Hodgepodge rules provide models where a variety of self-organizing structures develop.