

Some Experiments on Repeatability of Wolfram's Rule 30

By: Rich MacDonald
Email: rich@clevercaboose.com
Last Modified: 10/22/2004

Introduction

Everyone knows Wolfram's Rule30 as a basis for random number generation. It's also a useful "first-attempt" for someone getting their feet wet with cellular-automata programming. However, some issues immediately come to mind when we code it:

- 1) What do we do about the "endpoints" of the cell row? Does the row grow forever or 'wrap-around' as a ring.
- 2) How wide does the row need to be?

When I first started this document, I couldn't find the answer on the internet so I ran some tests myself. This document discusses the issues. When I was putting the document to bed I finally found the link to the relevant page in Wolfram's book. All can be found on p317, <http://www.wolframscience.com/nksonline/page-317>

Wolfram's answers are:

- 1) Use wrap-around
- 2) "A few hundred"

In essence, this document expands on a page of Wolfram's book p260, found at <http://www.wolframscience.com/nksonline/page-260>. However, I do go a step further in other sections.

What do we do about the "endpoints" of the cell row?

I can think of three different ways to handle the endpoints:

- 1) "Wrap-around" as a ring, i.e., where the first cell is considered to be the neighbor of the last. Some people call this a cylinder.
- 2) Never-ending expansion, i.e., where the row expands by 2 ever step. Wolfram appears to use this option in his drawings of the rule evolution¹, however, this is impractical for a random-number generator, since each point is O(the number of previous points), plus a significant RAM requirement. If I need a million random numbers, the million'th point will have a row size of a million and two.
- 3) Fix the endpoints to the same value, e.g., false or blank.

¹ In the Figure caption on <http://www.wolframscience.com/nksonline/page-29>, Wolfram states: "The pattern produced continues to expand on both left and right, but only the part that fits across the page is shown here." By this I infer that he generates the full expanding width. Also, his pictures show that he is definitely not using wrap-around.

It turns out that (3) is awful in terms of its periodicity. The generated pattern may be random (I never bothered to check) but it has a very low “period”. If any cell row is ever identical to one of the previous cell rows, thereafter the whole system will repeat forever. The generator may have wonderful randomness properties, but it may have a low period, and this is something that doesn’t seem to have been discussed in detail.

In the following, option 1 was used to deal with the endpoints.

Periodicity

I decided to run a simple test. Rule 30 is used for the cell generation. The starting row is all cells white(false) except for a single cell that is black (true), i.e., the standard starting point for the pictures in Wolfram’s work (<http://www.wolframscience.com/nksonline/page-27>). The endpoints are handled via wrap-around (option 1). For a given Cell Width, we generate new rows until a row is repeated. This was run for increasing widths until I ran out of RAM.

Table 1 lists the results. For a given width, the *count* is the generation step at which a row repeated. The *period* is the difference between the count and the first occurrence of the repeated row. In other words, a random number generator based on this approach will initially generate a non-repeating succession of *count* values, then it will start repeating every *period* values.

Table 1
Repetition Period of Rule 30 with wrap-around as a function of Cell Width

Width	Count	Period
3	3	1
4	8	8
5	6	5
6	9	1
7	4	4
8	41	40
9	73	72
10	22	15
11	177	154
12	126	102
13	272	260
14	1,428	1,428
15	1,502	1,455
16	6,074	6,016
17	10,902	10,846
18	3,569	2,844
19	553	247
20	3,912	3,420
21	2,317	597
22	4,481	3,256
23	39,410	38,249
24	187,063	185,040
25	590,313	588,425

26	313,243	312,156
27	241,384	240,300
28	267,813	249,165
29	845,018	833,808
30	387,569	374,265
31	2,990,087	2,841,150
32	871,811	842,528
33	1,192,298	1,049,268
34	5,831,219	5,656,002
35	18,636,338	18,480,630
36	197,368	2,844

The results are shocking. Periodicity is very low. For example, if we used a width of 36, our random number generator would repeat every 2,844 values.

Please note that I did not intentionally bias the table, given that the final row has such a horrible result. It was simply as far as my 2MB RAM computer could go. I was able to confirm that the count for width=37 is something greater than 21,000,000.

Figure 1 plots the *period* as a function of *width*. Overall the *period* increases with *width*, but not smoothly. Compare this plot with the plot shown in Wolfram p260, found at <http://www.wolframscience.com/nksonline/page-260>, the left-most plot on the 2nd row labeled “rule 30”. Interestingly his plot goes to cell width = 47 and a period of several billion. Wonder what kind of hardware he was able to get his hands on...

Figure 1
Repetition Period as a Function of Cell Width

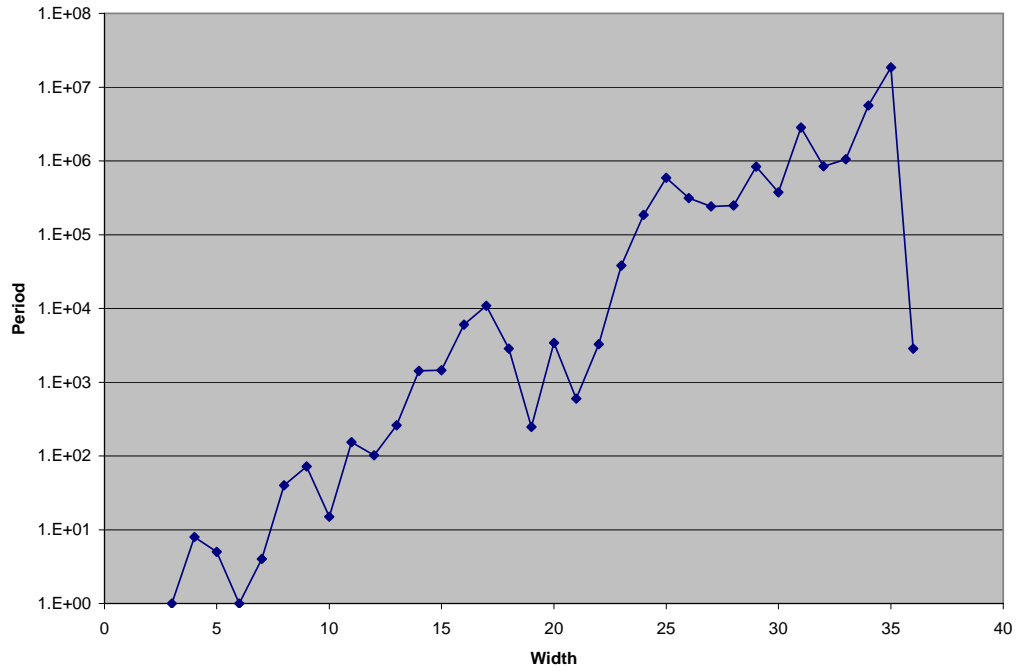


Figure 2 plots the ratio of the periods between any two widths, i.e., the increase in the period as a function of width; roughly, the derivative of the period as a function of width. Nothing really interesting here, just checking to see if the irregular nature of the plot is “random”; it looks ok.

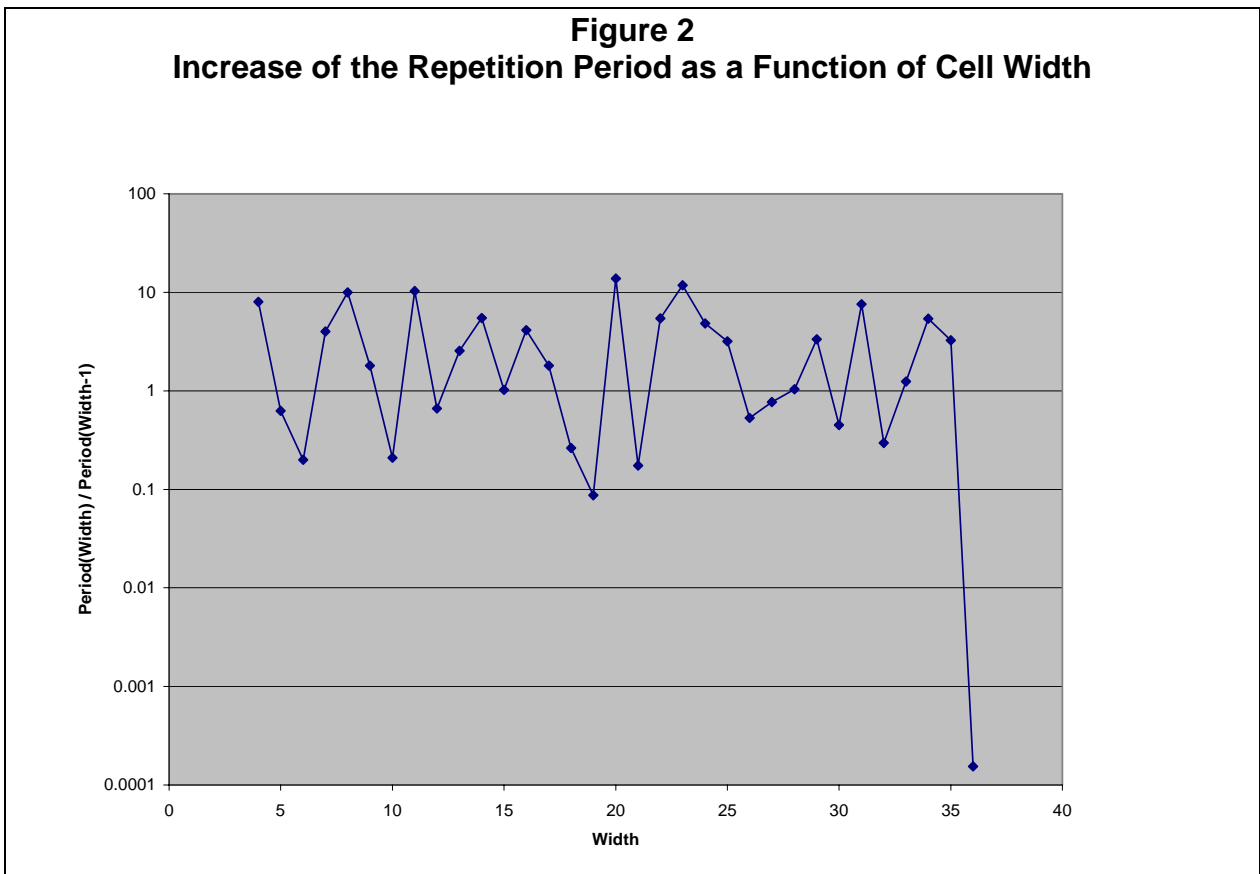
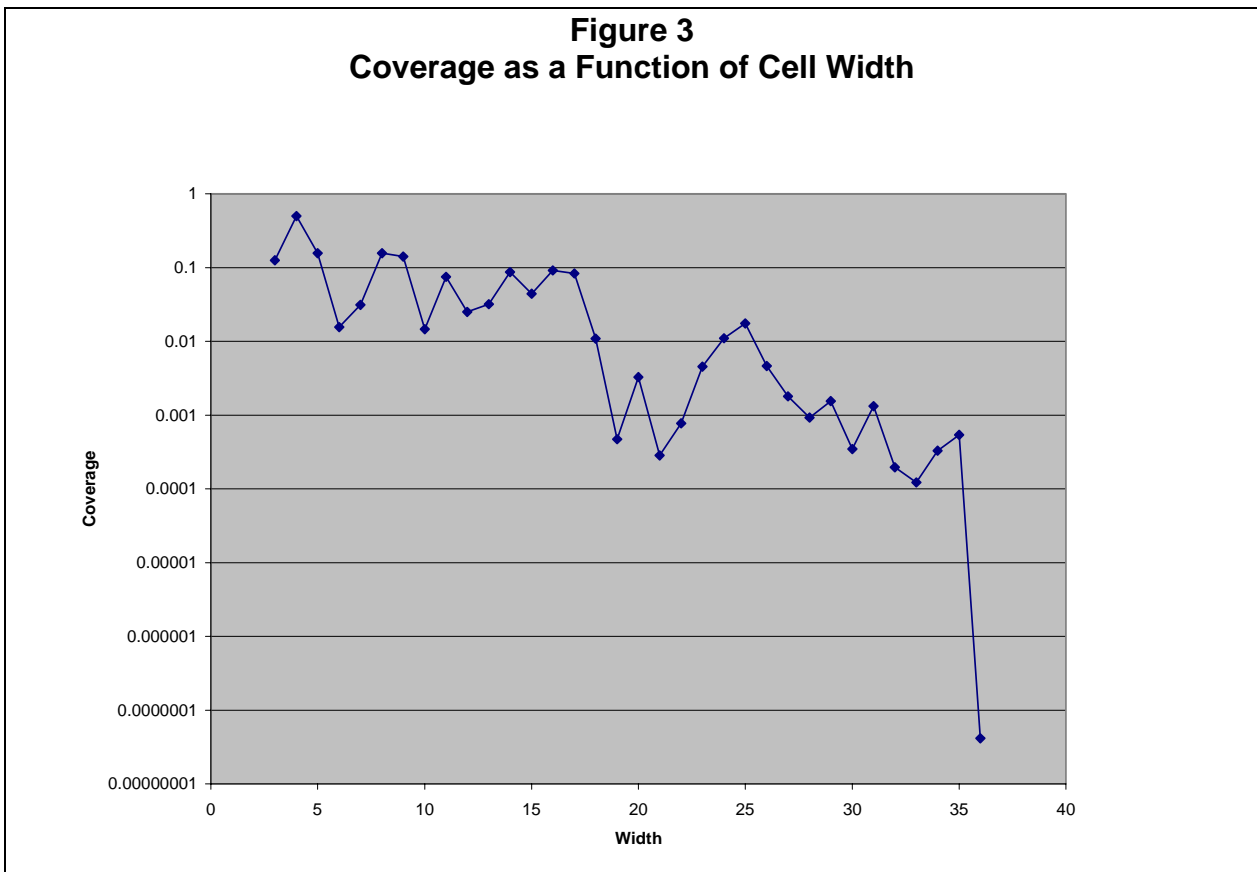


Figure 3 plots the *coverage* of the generator as a function of width. The *coverage* is the *count* divided by the “space” of the cells, i.e., 2^{width} . This shows that as width increases, coverage decreases.



All these results cast doubt on our random number generator. They indicate that we cannot use the approach with low widths. Or possibly the wrap-around approach is at fault. Now if our random number generator uses a width of 100, it's possible that this is a non-issue, i.e., the period may be so great that we never need to worry about it. However, Figure 3 indicates that we should not take this for granted, since coverage is likely to be infinitesimal at that point.

Unfortunately, it is hard to test this. With a 2GB RAM machine and width=37, I was able to generate 21,000,000 test points before hitting the wall. To go any higher, I have to switch to a RDBMS to check for repetition. I tried it, but this is orders-of-magnitude slower and I'd have to wait days to get the next point. No thanks. I've basically gone as far as I can go in this test. Plus I've (belatedly) found Wolfram's plot of the same data all the way up to width=47.

Fixing the endpoints to the same value

Instead of wrapping the endpoints of the cell row, I repeated the above test fixing the endpoints to false. Results were appalling, with periodicity no greater than 150 for width < 64. Obviously, this approach is not correct.

Expanding the endpoints at each step

I'd like to run the test where the endpoints are handled by expanding the row at each step. Unfortunately I don't know how to do it and what to conclude from the expected results. I know the approach is impractical in practice. Plus, there is no clear way to check for periodicity, since no row is ever the same as any of its predecessors. Probably this means that there is no such thing as repetition. We could arbitrarily pick some fixed range "inside" the row and check for repetition, but even if there is a match, the influences from outside this internal range will affect the future values of the internal range, so this proves nothing. I'm tempted to assert that no periodicity will be encountered, so I guess we can all breathe a sign of relief and grant Wolfram that his "New Kind of Science" still holds in an expanding universe ☺

Starting from Random Values

In the above tests, my initial row seed was to set all but one cell to zero/false. In practice, random number generators may seed the initial row randomly. Also, using random initial rows we can run this test many times to search for "average periodicity". In other words, my first test found one period, but within the actual "space" of the model there are likely to be many periods. Repeating the test with many initial random seeds allows us to estimate the size of these periods.

So I repeated the test with random initial row seeds. I used a Boolean random number generator based on rule 30 with width=100, endpoints as wrap-around, and all values but one of the initial row being false. This generator was restarted at each new width. I ran this test for a range of widths and repeated the test 100 times at each width. In other words, I picked a single width, seeded the Boolean random number generator, generated an initial cell row randomly, stepped the cell row until it repeated, then generated another initial cell row and repeated this many times. I did this for a series of widths. Of interest was the "average" value of the periodicity, which gives us a rough idea of the periodicity of the model for that width. Also of interest was the lowest value of the periodicity, to give us an idea of the worst-case. Note that this lowest value is not the true lowest value because I could only generate a finite number of tests and therefore did not completely cover the full space of the model.

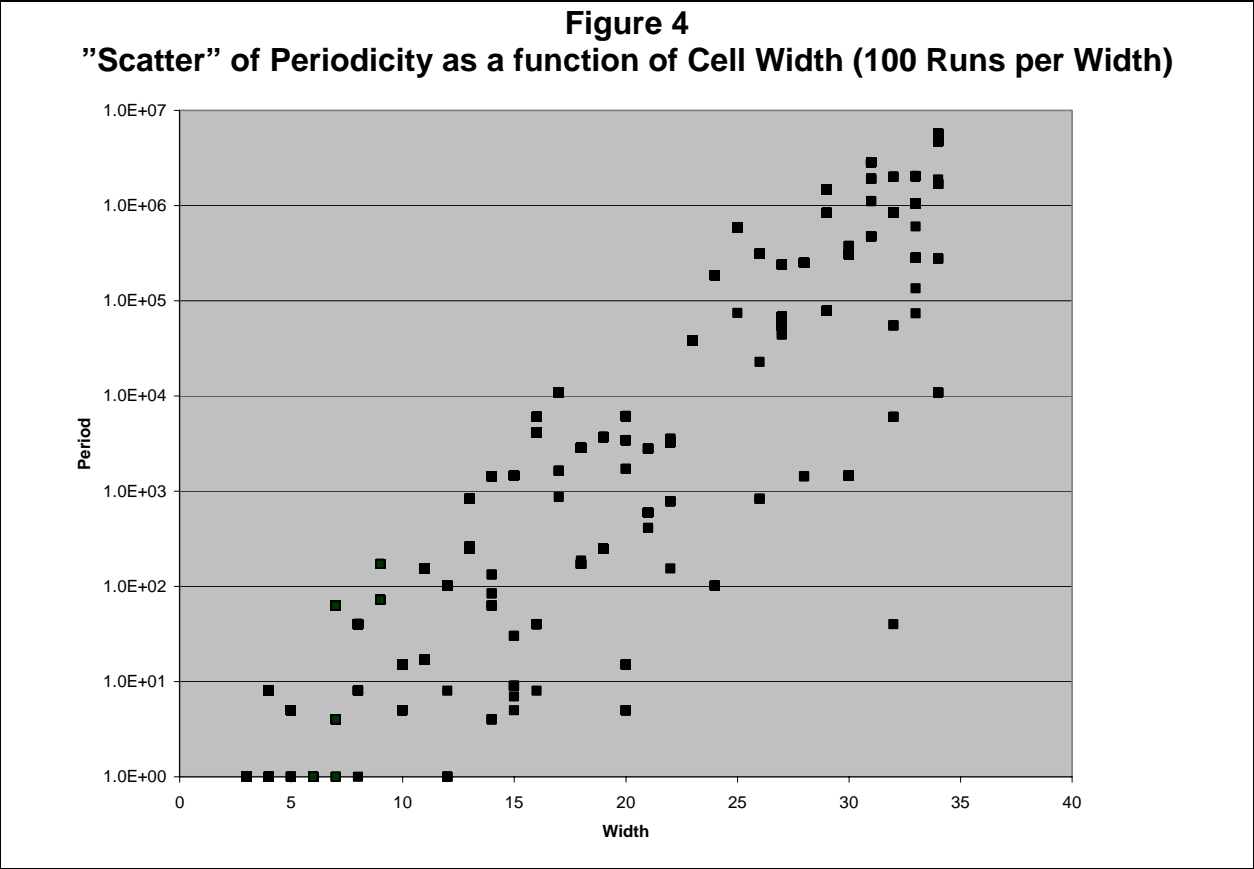


Figure 4 shows the results. In practice, far fewer than 100 periods were encountered (see Figure 5), but *all* 100 runs eventually ran into one of these periods.

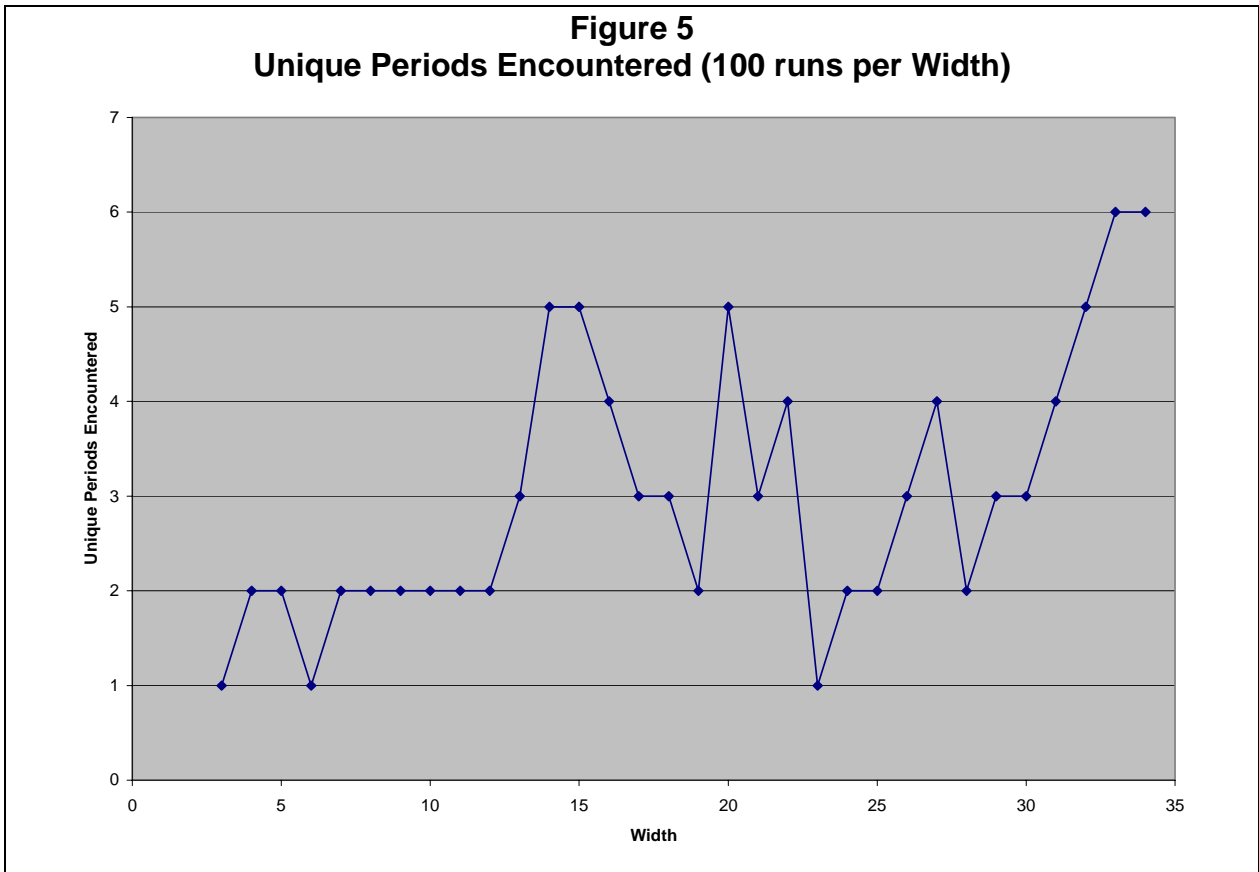
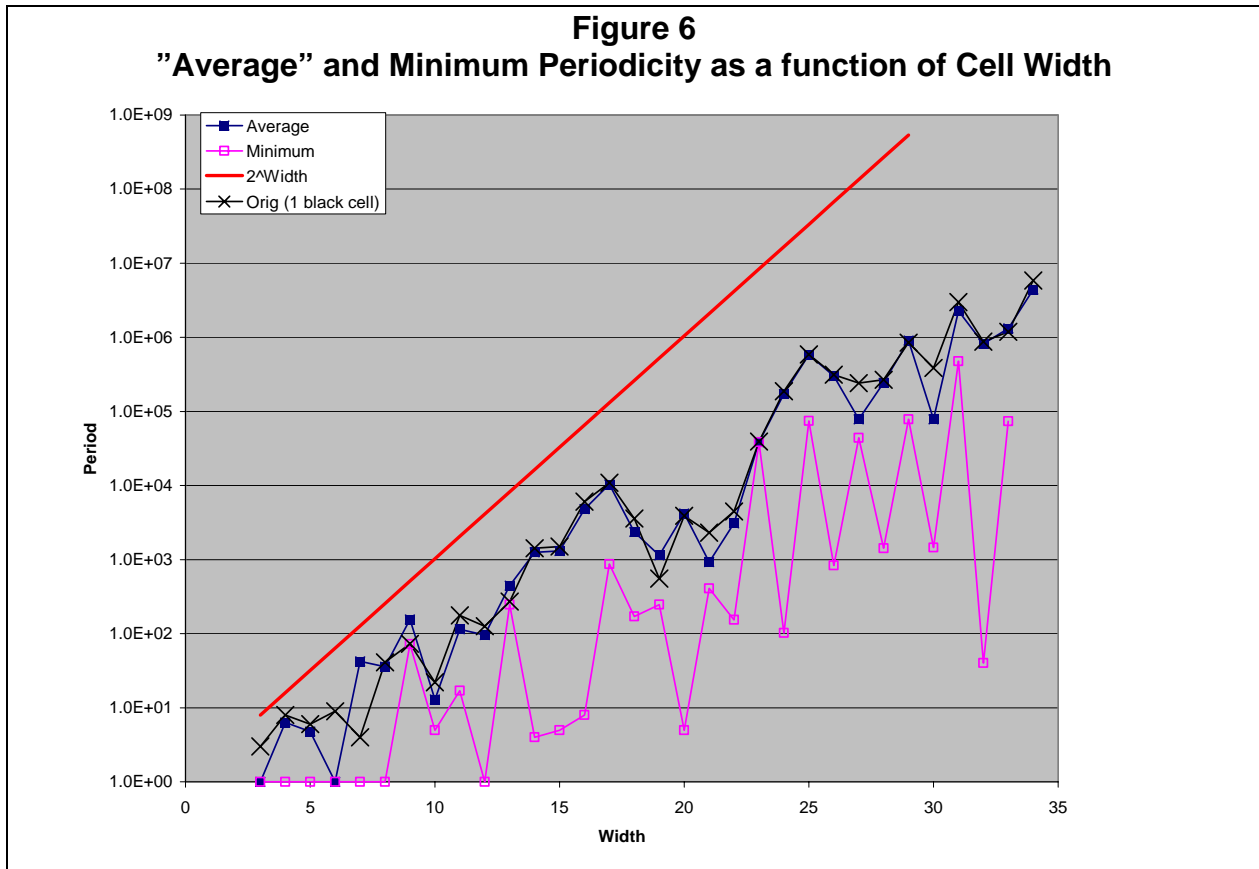


Figure 6 shows the same results as Figure 4 with the “average” (simply the mean of all the periods) and minimum periods. I have also plotted the “coverage”, which in this context can be interpreted as the period if there were no repetitions, i.e., the theoretical maximum period. Clearly, the “average” or “encountered” periodicity is much less than the theoretical maximum, and while the average is increasing with cell width, it may be *decreasingly* increasing. As Figure 4 shows, the greater the cell width the more repetition points we will encounter. Once again, however, we are far too far away from practical cell widths to extrapolate, but the trends are a cause for concern.



Eyeballing Wolfram's page 260

My Figure 1 compares with Wolfram's at <http://www.wolframscience.com/nksonline/page-260>. He is also able to extend the run much further than I. However, I think my idea of repeating the test with random starting points provides an additional check. Thus, on Figure 6 I have also plotted the results from the original run, i.e., with one black cell and the rest white. Note that this curve is practically on top of the average curve, meaning that the original test captured the 'average periodicity' rather well. This means that we can extrapolate Wolfram's figure as a good indicator of the average periodicity as well.

Future Work

More interesting things I guess, but I've milked this issue for more than its worth. Does anyone know a good method for converting random Boolean values to random, integers, longs, floats and doubles?

Conclusions

In a nutshell, rule 30 with end-point wrap-around has a far lower period of repetition than people may have realized. Not Wolfram, obviously as he already did this study ☺. Nevertheless, his single plot is easy to overlook and the numbers are unacceptable for anything less than a cell width of 45 or so.

It is also fair to say that there is a small (ok, *very* small) but finite possibility that some unlucky combinations will have very poor randomness properties. For example, who

would have thought that for a cell width = 33, there exists a period of only $40!^2$ Who is to say that this isn't also the case for cell width = "a few hundred". Ok Wolfram, I'll grant that you're probably correct when you say that "even with the fastest foreseeable computers, the actual period of repetition will typically be more than a billion billion times the age of the universe"³, but are you really really sure :-?

² Its there, take a look at Figure 4.

³ <http://www.wolframscience.com/nksonline/page-317>