THE CELLULAR AUTOMATON PULSING MODEL, EXPERIMENTS WITH DDLab[*]

Andrew Wuensche

Discrete Dynamics Lab, United Kingdom
http://www.ddlab.org

Edward Coxon

Department of Anaesthesia and Pain Medicine
The Canberra Hospital, ACT, Australia

(Received December 7, 2018)

The cellular automaton (CA) pulsing model described the surprising phenomenon of spontaneous, sustained and robust rhythmic oscillations, pulsing dynamics, when random wiring is applied to a 2d "glider rule" running in a 3-value totalistic CA. Case studies, pulsing measures, possible mechanisms, and implications for oscillatory networks in biology were presented. In this paper, we summarise the results, extend the entropy-density and density-return map plots to include a linked history, look at totalistic glider rules with neighbourhoods of 3, 4 and 5, as well as 6 and 7 studied previously, introduce methods to automatically recognise the wave-length, and extend results for randomly asynchronous updating. We show how the model is implemented in DDLab to validate results, output data, and allow experiments and research by others.

DOI:10.5506/APhysPolBSupp.12.135

1. Introduction

The cellular automaton pulsing model (the CAP model) [1] is a 2d cellular automaton (CA) subject to a 3-value k-totalistic "glider rule", where the local wiring is randomised. Pulsing dynamics, sustained rhythmic oscillations of density and entropy measures, emerge spontaneously and almost inevitably. The characteristic wave-forms are robust and depend on the

[*] Presented at the Summer Solstice 2018 Conference on Discrete Models of Complex Systems, Gdańsk, Poland, June 25–27, 2018.

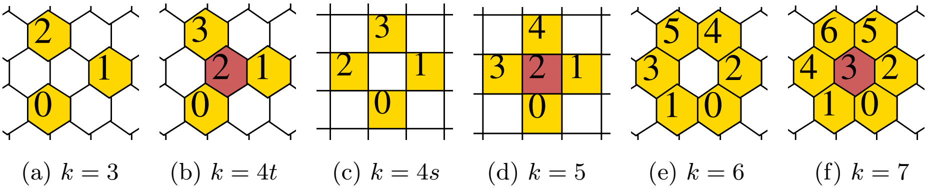
specific glider rule applied. If the extent or reach of random wiring is reduced, pulsing will eventually break down, suggesting a threshold and phase transition.

The CAP model is a significant phenomena in its own right, posing questions in CA theory on the mechanisms of glider dynamics, the mechanisms of pulsing, and how the two are related. It is also significant in the context of bio-oscillations ubiquitous at many time/size scales in biology. Currently, there is no satisfactory theory to explain essential oscillations in various animal organs, for example heart beat, uterine contractions in childbirth, and various rhythmic behaviours such as breathing controlled by the central nervous system. The CAP model provides a possible oscillatory biomechanism based on long-range signalling between cells/neurons in excitable tissue, following the classical three state dynamic: Firing, Refractory, and Ready-to-fire, and subject to glider rule equivalent logic.

These concepts, ideas and results were defined and documented in [1]. In this paper, we present a summary, extend the entropy-density and density-return plots to include a linked history, and look at glider rules with smaller neighbourhoods k, of 3, 4 and 5, as well as 6 and 7 studied previously. We introduce methods to automatically recognise and measure wave-length/wave-height and output the data. We extend results for randomly asynchronous updating. We show how the model is implemented in DDLab to allow validation of results and further experiments and research by others, and include pre-assembled collections of glider rules.

2. Glider dynamics in 2d CA

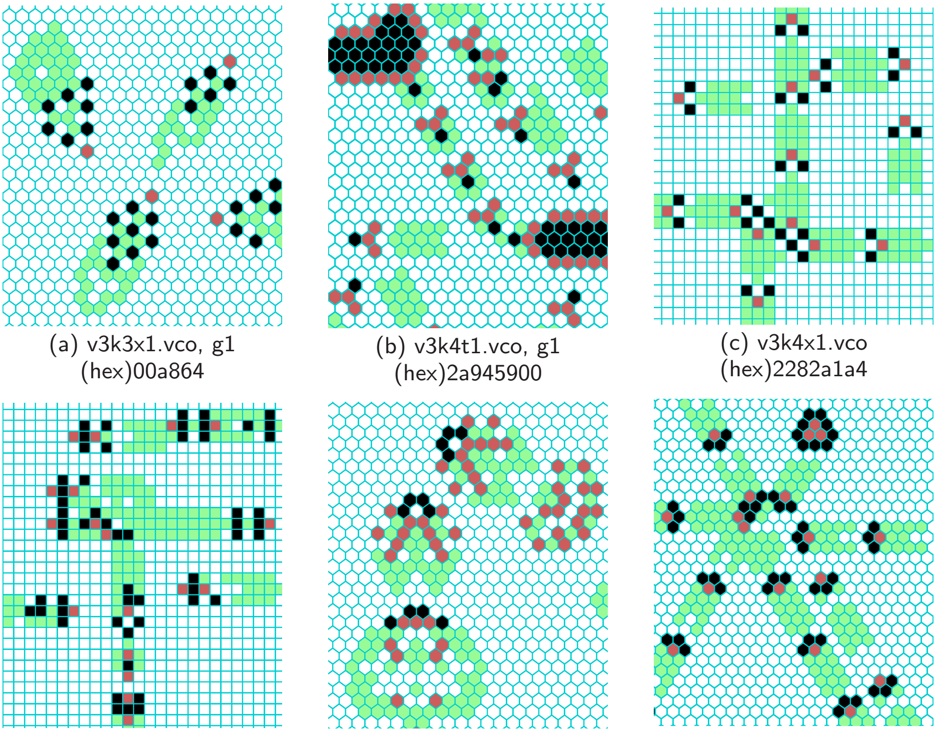
The lattice geometry of a 2d CA depends on its neighbourhood template, and we present templates relevant to this paper in Fig. 1. Collections of glider rules have been assembled based on these templates.



(a) $k = 3$ (b) $k = 4t$ (c) $k = 4s$ (d) $k = 5$ (e) $k = 6$ (f) $k = 7$

Fig. 1. 2d neighbourhood templates (k = 3 to 7) as defined (and numbered) in DDLab, setting the lattice geometry, both hexagonal and square. The target cell is central even if not part of the template.

Each cell in the lattice updates its value synchronously[1] according to a homogeneous 3-value k-totalistic rule. This determines the dynamics which can be seen as successive pattern images in the same way as a series of still images make a movie. Most rules result in disorder, but we are interested in complex rules characterised by identifiable mobile features, in particular gliders or mobile particles consisting of compact cell-value assemblies moving through the lattice with a given velocity, comprising a head and tail, and interacting by collisions with other gliders or stable particles as in the examples in Fig. 2.



(a) v3k3x1.vco, g1
(hex)00a864

(b) v3k4t1.vco, g1
(hex)2a945900

(c) v3k4x1.vco
(hex)2282a1a4

(d) v3k5x1.vco, g1
(hex)004a8a2a8254

(e) v3k6n6.vco, g16
(hex)01059059560040

(f) v3k7w1.vco, g1
(hex)020609a2982a68aa64
The Spiral rule [2]

Fig. 2. Examples of glider dynamics for 2d neighbourhood templates ($k = 3$ to 7) in Fig. 1. Cell values: 0 = white, 1 = grey/red, 2 = black. Light grey/green trails of 5 time-steps indicate glider velocity. Examples (b), (c), (e), and (f) include glider-guns. The rules can be loaded in DDLab by their filenames, in hexadecimal, or from the rule collections index $g(x)$.

[1] Although synchronous updating is a necessary condition for gliders to emerge, pulsing in the CAP model persists for asynchronous and noisy updating [1].

In DDLab, collections of glider rules are provided, extracted from automatic samples of complex rules — not all complex rules support gliders and pulsing. The older collections for $k = 6$ and 7 relating to [1] include complex rules as well as pulsing rules, whereas for the $k = 3$, 4, and 5 collections, only pulsing rules have been included. For $k = 3$, pulsing results from mobile intersecting linear structures, rather than classic gliders which are less frequent.

While assembling these collections, we can confirm that gliders almost inevitably imply pulsing, and their absence imply non-pulsing. We should, however, note that we have observed a few very rare exceptions.

3. Random wiring

Unrestricted, unbiased, random wiring follows the same connection approach as Kauffmans's "Random Boolean Networks" [2]. For each target cell, we take k cells at random anywhere in the lattice and "wire" them to distinct cells in the pseudo-neighbourhood template — "pseudo" because the actual template values are replaced by the values of the random cells, as in Fig. 3 (a). Each target cell is assigned its own random wiring. We have also studied random wiring with three types of bias resulting in degrees of degraded pulsing and waveform signatures [1]. A summary of the consequences relating to Fig. 3 (a), (b), (c), (d) are listed below,

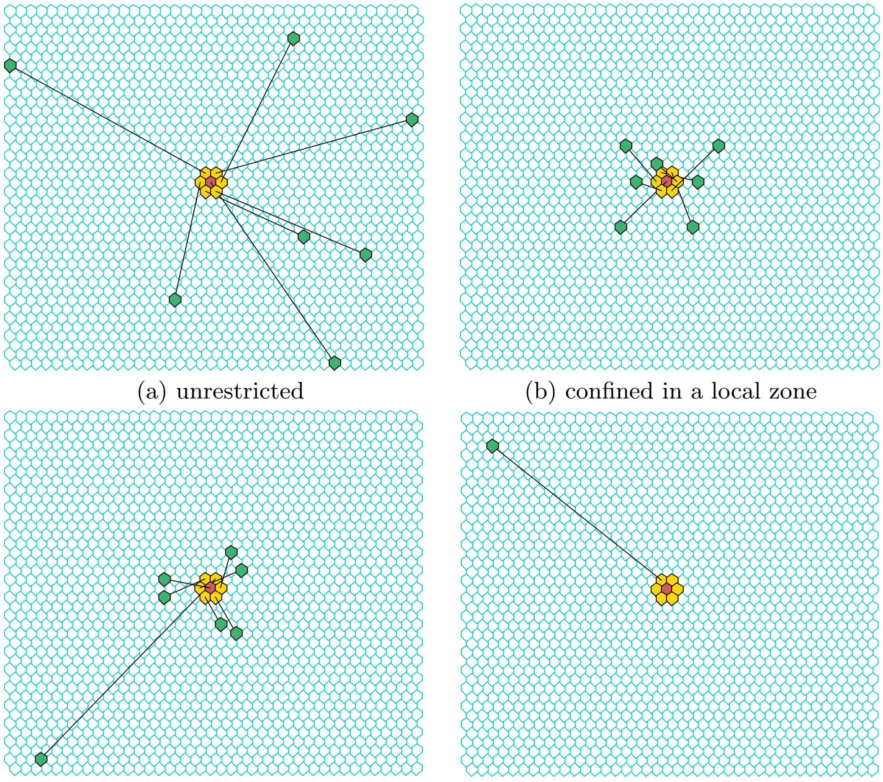
(a) Unbiased random wiring gives the strongest and most robust pulsing in the CAP model.

(b) If random wiring is confined within a local zone, as the size of the zone is reduced, at some stage, global pulsing will break down into mobile patches of density which may take the form of spiral waves. The threshold properties of this phase transition requires further investigation. The waveform signatures are recognisable as they degrade.

(c) If one wire is "freed" from a small local zone, robust pulsing with a characteristic waveform is partially restored, and this is reinforced as a higher proportion of wires are freed.

(d) Starting with a conventional CA, freeing one or more wires randomly from each neighbourhood template results in the onset of pulsing by degrees, also conforming to a characteristic waveform.

In these experiments, the precise random wiring, the random initial state, and even re-randomising the wiring at each time-step as in Derrida's annealed model [3], makes no significant difference to the waveform.



(a) unrestricted

(b) confined in a local zone

(c) confined but one wire freed

(d) CA but one wire freed

Fig. 3. Examples of random wiring in a 40 × 40 2d lattice. A bigger lattice would normally be required for robust pulsing, to avoid reaching a uniform value attractor [4].

A single key press in DDLab enables switching between CA and any type of preset random wiring, or between stable random wiring and re-randomising the wiring at each time-step while maintaining the preset bias.

4. Why 3-value k-totalistic rules?

We restrict our investigation to the subset of 3-value k-totalistic rules for the following reasons:

— The discovery of pulsing in the CAP model, and that no pulsing is evident in an equivalent 2-value system.

— Compared to a general CA, the k-totalistic rule-table is relatively short [5, Sec. 13.6.1] and thus tractable for displaying aspects of the pulsing waveform.

- The dynamics are isotropic so closer to nature — the same output for neighbourhood template rotation or reflection, though k-totalistic rules are restricted beyond isotropy giving smaller rule-spaces than just isotropic rules.
- The rules can be reinterpreted as reaction-diffusion systems with inhibitor/activator reagents in a chemical medium [6–8], where the three CA values are seen as: Activator, Inhibitor, and Substrate.
- The availability of short-lists of glider rules, extracted from large samples of complex rules that are found (and sorted automatically) by the variability of input-entropy [5, 9, 10].
- The CAP model can be applied to bio-oscillations in excitable tissue according to classical 3-state neuronal dynamics: Firing, Refractory, and Ready-to-fire.

5. Definition of 3-value k-totalistic rules

The properties and definition of 3-value k-totalistic rules are summarised as follows:

- The target cell at time step t depends on the combination of k totals, or frequencies, of the values in the neighbourhood template at $t-1$.
- Each combination of totals make up the rule-table (named “kcode”), for example, the 3-value ($v=3$) $k=5$ rule v3k5x1.vco in Fig. 2 (d),

```
               20...................0 <--kcode index
                |                   |
           > 2: 544333222211111000000  < frequency strings       5    0
v=3 values > 1: 010210321043210543210  < of 2s, 1s, 0s,     from 0 to 0
           > 0: 001012012301234012345  < shown vertically        0    5
                |||||||||||||||||||||
                010222022022220021110  <--rule table (kcode), outputs [0,1,2]
```

  In DDLab, kcode can be expressed in hexadecimal for compactness, in this case 004a8a2a8254, also shown in Fig. 2 (d).

- kcode size $S=(v+k-1)!/(k!\times(v-1)!)$, which increases arithmetically; kcode-space $= v^S$. A general rule-table (rcode) has v^k entries increasing exponentially; rcode-space $= v^{v^k}$. For $v=3$ and $k=3$ to 7, the size of the kcode and rcode strings are as follows:

k	3	4	5	6	7
vcode	10	15	21	28	36
rcode	27	64	125	216	343

6. Signs of pulsing

When pulsing occurs, it is clearly evident by eye when the system is run in DDLab (Fig. 4). Space-time pattern density will exhibit a steady rhythmic periodic beat, and this will be reflected by the input-histogram. The input-histogram tracks how frequently the different entries in a rule-table are actually looked up at each time-step, or the changing 2d block-frequency, where the "blocks" are the alternative patterns within the neighbourhood template.

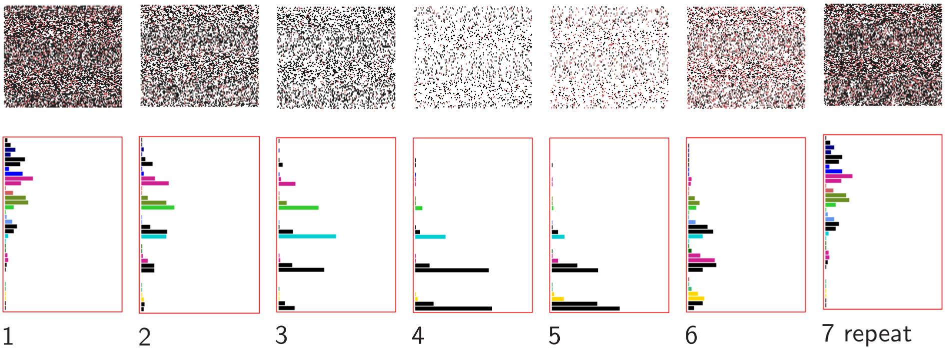


Fig. 4. Dynamic graphics in DDLab show up pulsing in the v3k7 Spiral rule [6] from Fig. 2 (f), with a period of 7 time-steps. Each 100 × 100 pattern in a typical cycle is shown above its input-histogram where horizontal bars represent the lookup-frequency of 36 neighbourhoods, (all-2s at the top) for each corresponding time-step.

Further time-plot measures can be derived and presented graphically as the system iterates. The entropy of the input-histogram can be calculated[2] and plotted with its characteristic wave-length (wl), wave-height (wh, twice amplitude), and waveform (its shape or phase), which in turn can generate an entropy-density scatter plot [9] (Fig. 5). From space-time patterns, the density or proportion of each value, (0, 1, 2) if $v = 3$, can be plotted, and this can generate a density return-map scatter plot [5] (Fig. 6). The scatter plots have the characteristics of chaotic strange attractors, and successive dots can be connected to create a linked history — this option is much faster to produce the characteristic plot because just a few time-steps are needed. The wl and wh data can be recognised and output automatically.

[2] The input-entropy is the Shannon entropy H of the input-histogram. For one time-step, $H^t = -\sum_{i=0}^{S-1}\left(Q_i^t/n \times \log_2\left(Q_i^t/n\right)\right)$, where Q_i^t is the lookup-frequency of neighbourhood i at time t. S is the rule-table size, and n is the network size. The normalised Shannon entropy H_{N} is a value between 0 and 1, $H_{\mathrm{N}} = H^t / \log_2 n$, which measures the heterogeneity of the histogram — "entropy" in this paper refers to H_{N}.

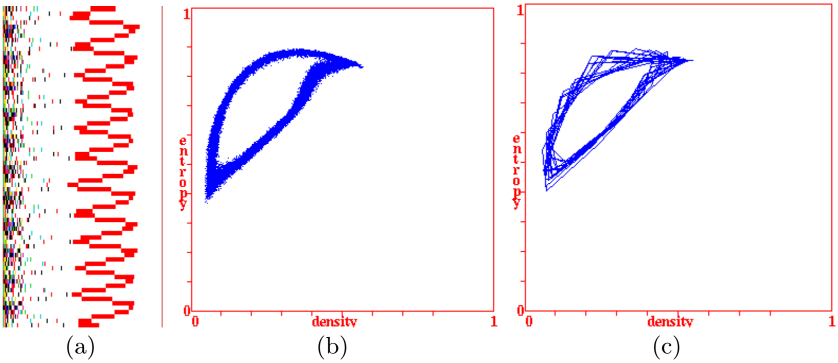


Fig. 5. For the v3k7 Spiral rule from Fig. 2 (f), (a) Input-entropy oscillations with time (y-axis, stretched), wl ≈ 7 time-steps, wh ≈ 0.4. Left edge: superimposed histogram values plots. (b) The entropy-density scatter plot — input-entropy (y-axis) against the non-zero density (x-axis), for about 33 000 time-steps. (c) The same plot for just a few pulsing cycles, but linking successive dots giving a time-history.

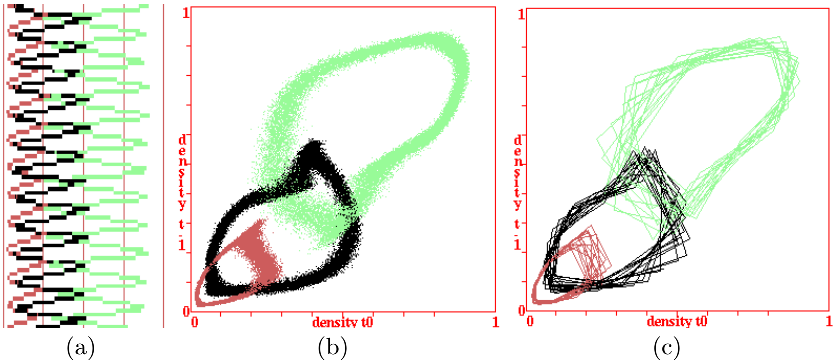


Fig. 6. For the v3k7 Spiral rule from Fig. 2 (f), (a) Value-density oscillations with time (y-axis, stretched). 0 = light grey/green, 1 = grey/brown, 2 = black. (b) The density return-map scatter plot — the density of each value at t_0 (x-axis) against its density at t_1, plotted as coloured dots as above for about 33 000 time-steps. (c) The same plot for just a few pulsing cycles, but linking successive dots giving a time-history.

We will use the term "waveform" to sum up these pulsing measures. Each glider rule in the CAP model maintains its distinctive waveform signature, reflecting the distinctive glider dynamics. It was shown in [1] that the underlying waveform signature is independent of the network size n, becoming more focused as n increases towards infinity, but reducing n makes reaching a uniform attractor [4] more likely, where the system would freeze.

Waveform measures and plots are usually averaged over a moving window of w time-steps, $w = 10$ to classify rules by the variability of input-entropy [9], but to observe pulsing dynamics most effectively, we take the measures over each time-step, where $w = 1$. However, when measuring the wave-length wl automatically, $w \geq 2$ can sometimes be preferable. Figures 5 and 6 show examples of the measures for the $k = 7$ Spiral rule [6] on a 100×100 lattice, with $w = 1$.

7. Wave-length and wave-height data

A new method is available in DDLab to automatically recognise and measure the wave-length (wl) and wave-height (wh) of entropy oscillations in the CAP model. The output appears in the terminal. The data can be activated while the density-entropy plot is active. The algorithm is effective for well-developed steady (but possibly variable) entropy oscillations — the examples in Fig. 7 relate to the rules in Figs. 9, 10, and 12. Entropy oscillations with jagged stretches or transient min/max values in the plot profile, as in Figs. 8 and 12 (a), can disturb the wave-length (wl) measures, but this can be smoothed out by making the time-step window $w \geq 2$ without effecting wl, though wh would be reduced. Figure 8 gives an example of variable wl, and with jagged stretches resolved by making $w = 20$. Section 10.2 includes step-by-step instructions for the method.

```
v3k3 kcodeSize=10
(hex)00a864 w=1
min=112 wl=10 wh==0.268
max=115 wl=9
min=121 wl=9 wh==0.260
max=125 wl=10
min=131 wl=10 wh==0.269
max=135 wl=10
min=140 wl=9 wh==0.211
max=144 wl=9
```

(a) $w = 1$, Fig. 9

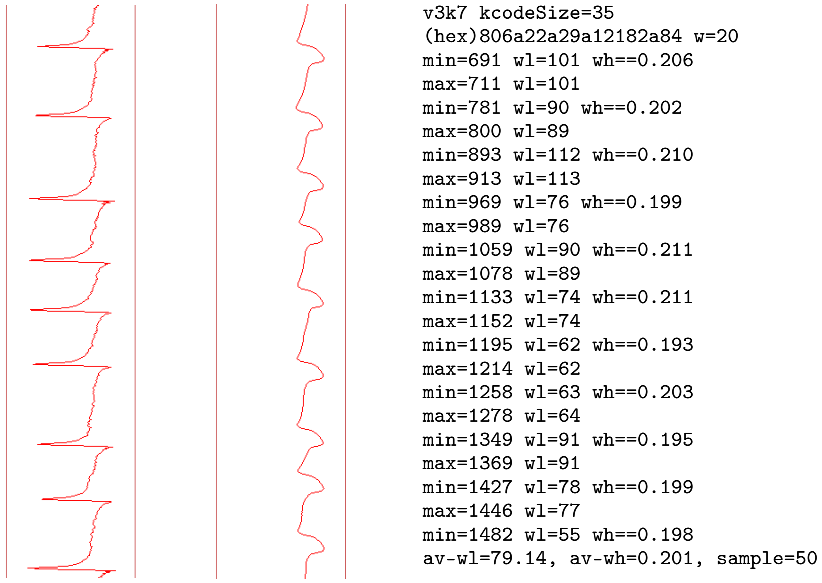
```
v3k4 kcodeSize=15
(hex)2a945900 w=1
min=107 wl=6 wh==0.294
max=110 wl=7
min=114 wl=7 wh==0.332
max=117 wl=7
min=121 wl=7 wh==0.270
max=123 wl=6
min=127 wl=6 wh==0.355
max=129 wl=6
```

(b) $w = 1$, Fig. 10

```
v3k5 kcodeSize=20
(hex)004a8a2a8254 w=2
min=116 wl=11 wh==0.428
max=118 wl=11
min=126 wl=10 wh==0.420
max=129 wl=11
min=138 wl=12 wh==0.457
max=141 wl=12
min=148 wl=10 wh==0.442
max=151 wl=10
```

(c) $w = 2$, Fig. 12

Fig. 7. Wave-length (wl) and wave-height (wh) examples showing data for a typical sequence of 4 pulsing cycles. Data is output continuously in the terminal, and average values so far if interrupted. The rules, shown at the top, relate to waveform figures indicated. The algorithm identifies the time-step at the minimum and maximum values of each oscillation to calculate wl and wh. The size of the time-step window w is shown; usually $w = 1$, but, for example (c) $w = 2$ to smooth out a jagged stretch at the maximum part of the plot profile in Fig. 12 (a).



```
v3k7 kcodeSize=35
(hex)806a22a29a12182a84 w=20
min=691 wl=101 wh==0.206
max=711 wl=101
min=781 wl=90 wh==0.202
max=800 wl=89
min=893 wl=112 wh==0.210
max=913 wl=113
min=969 wl=76 wh==0.199
max=989 wl=76
min=1059 wl=90 wh==0.211
max=1078 wl=89
min=1133 wl=74 wh==0.211
max=1152 wl=74
min=1195 wl=62 wh==0.193
max=1214 wl=62
min=1258 wl=63 wh==0.203
max=1278 wl=64
min=1349 wl=91 wh==0.195
max=1369 wl=91
min=1427 wl=78 wh==0.199
max=1446 wl=77
min=1482 wl=55 wh==0.198
av-wl=79.14, av-wh=0.201, sample=50
```

(a) $w = 1$ (b) $w = 20$ (c) $w = 20$

Fig. 8. Wave-length (wl) for rule v3k7 $g35$ in [1], with steady but variable wl oscillations, and jagged stretches on the downslope of the entropy plot profile (a) can give false min/max results, but this is fixed by increasing the time-step window ($w = 20$) to smooth the plot (b). (c) shows typical data, with wl between 62 and 113 time-steps, though the actual range is slightly greater. The last line shows average values.

8. CAP model plots for $k = 3, 4, 5$, and 6

The CAP model input-entropy and value-density plots, as in Figs. 5 and 6, for the v3k7 Spiral rule from Fig. 2(f), on a 100 × 100 lattice, are shown here for the rules in Section 2, for $k = 3$, $k = 4$ (triangular and orthogonal), $k = 5$, and $k = 6$. For each rule, four plots (a), (b), (c), (d) described below, are shown in Figs. 9–13. The rules can be loaded in DDLab in various ways including by their filenames or in hexadecimal, but to explore the range of pulsing behaviours most effectively, from the rule collections index $g(x)$.

(a) Input-entropy oscillations with time (y-axis, stretched). Left edge: superimposed histogram values plots.

(b) The entropy-density scatter plot — input-entropy (y-axis) against the non-zero density (x-axis), for just a few pulsing cycles, and linking successive dots giving a time-history.

(c) Value-density oscillations with time (y-axis, stretched). 0 = light grey/green, 1 = grey/brown, 2 = black.

(d) The density return-map scatter plot — the density of each value at t (x-axis) against its density at $t+1$, plotted as coloured dots for just a few pulsing cycles, and linking successive dots giving a time-history.

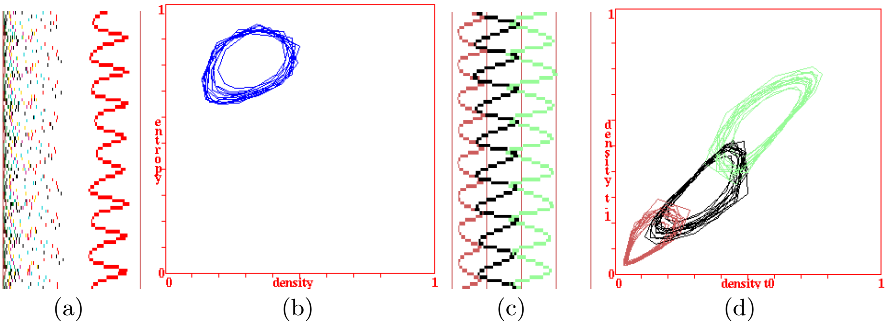


Fig. 9. $k3$ CAP plots, v3k3x1.vco, (hex)00a864, $g(1)$.

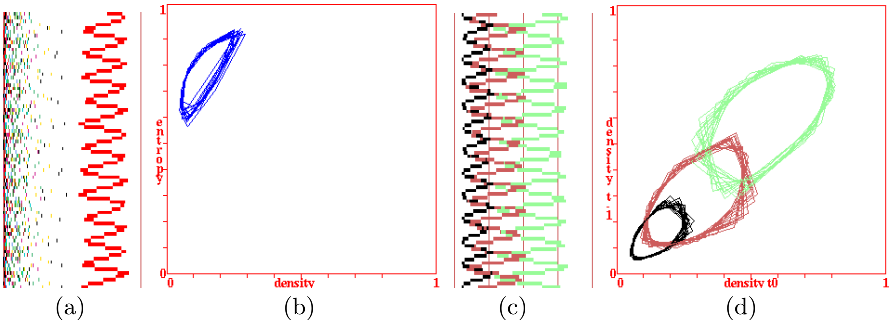


Fig. 10. $k4$ (triangular) CAP plots, v3k4t1.vco, (hex)2a945900, $g(1)$.

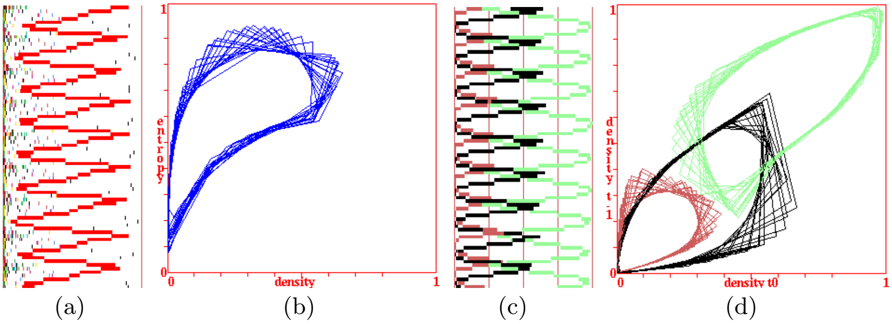
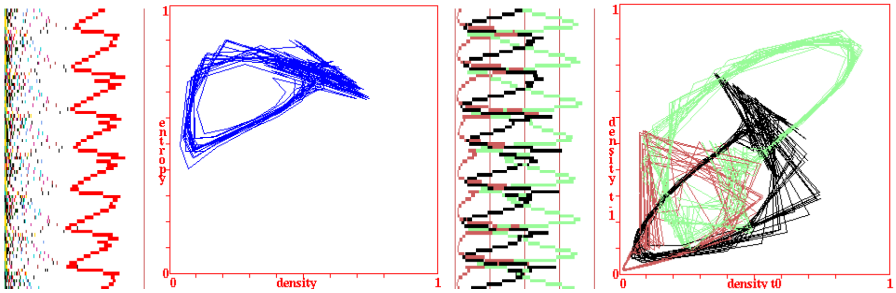
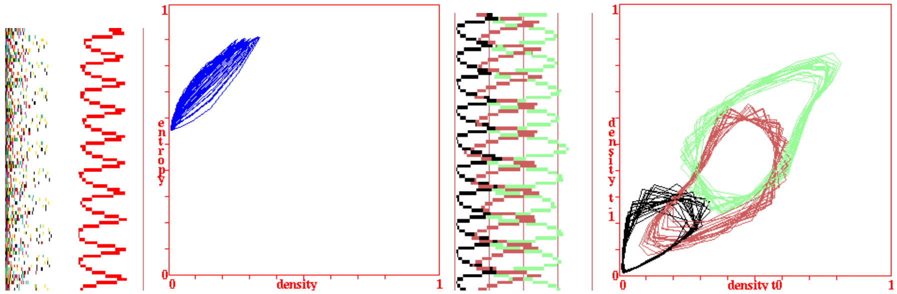


Fig. 11. $k4$ (orthogonal) CAP plots, v3k4x1.vco, (hex)2282a1a4.



(a) (b) (c) (d)

Fig. 12. $k5$ CAP plots, v3k5x1.vco, (hex)004a8a2a8254, $g(1)$.



(a) (b) (c) (d)

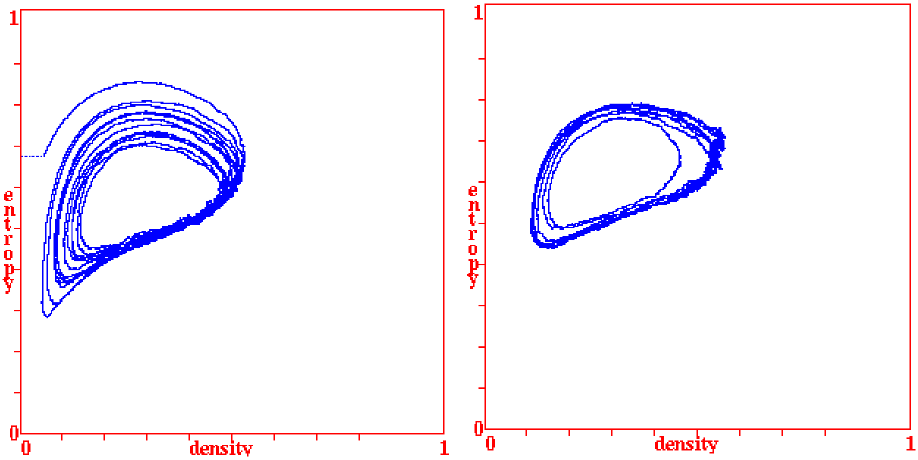
Fig. 13. $k6$ CAP plot, v3k6n6.vco, (hex)01059059560040, $g(16)$.

9. Randomly asynchronous updating

Pulsing in the CAP model subject to asynchronous and noisy updating turns out to be robust [1], but perhaps the most intriguing and unexpected result is that pulsing continues with a recognisable waveform when random single cells are updated one at a time. Experiments for $k = 6$ and $k = 7$ glider rules[3] (Fig. 14) confirm these results, where data is plotted for each cell update so experiments take 10 000 times longer (for 100×100) than synchronous updating. To seed up computation, the data can be sampled at longer intervals by partial order or sequential updating, or a combination of both.

Although the CAP model is a computer simulation, the fact that the pulsing waveform is preserved for randomly sequential singe cell updating is significant in the sense that the CPU timer can be ruled out as an external time-keeper.

[3] For other rules and k values, the results are still under investigation.



(a) v3k7 $g3$ wl ≈ 21, wh ≈ 0.6 (b) v3k7 $g35$ wl, wh, see Fig. 8

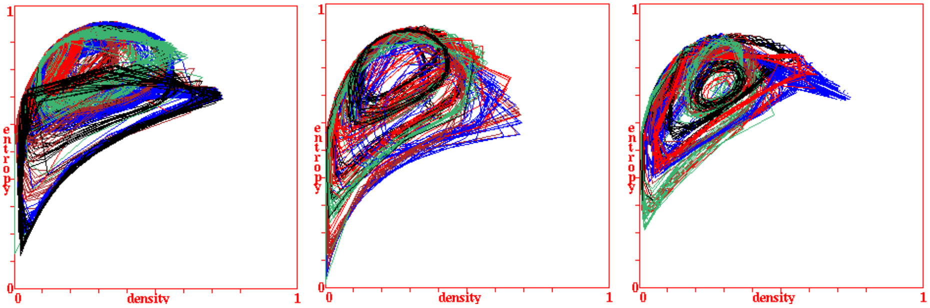
Fig. 14. Entropy-density plots for sequential updating one cell at a time at random positions [1] result in pulsing. The network is 100×100, so $n = 10\,000$ such updates are required to approximate one synchronous time-step. (a) v3k7 $g3$ (see Fig. 16), and (b) v3k7 $g35$ (see Fig. 8).

10. Experiments with DDLab

Using the DDLab software [11], the results presented in [1] and in this paper can be reproduced, and many other rules and aspects of pulsing dynamics investigated. Pre-assembled collections of glider rules are available, and can be activated on-the-fly (key **g**), while space-time patterns are active, the wiring can be toggled between CA and random (key **7**), and the dynamics observed, measured and recorded with other keys and interactive functions. The older collections for $k = 6$ and 7 relating to [1] include complex rules as well as pulsing rules. For the $k = 3$, 4, and 5 collections, only pulsing rules have been included — figure 15 shows an overlay of all 20 entropy-density plots for each of these rule collections.

We summarise below the steps in DDLab to run the experiments, referring to chapters and sections (denoted by #x.x) from the book "Exploring Discrete Dynamics — Second Edition" [5], — its pdf is kept updated online [11]. Having installed DDLab[4], from a terminal in the directory `ddlab/ddfiles`, enter **../ddlabz07 -w &** to start in a white screen.

[4] Download the latest compiled version of DDLab (November 2018 or later) for Linux or Mac from `www.ddlab.org` to a directory called `ddlab`, and the extra files in `dd_extra.tar.gz` to a subdirectory called `ddlab/ddfiles` (directory names are arbitrary). #3 gives further guidance. For Microsoft Windows, use a Linux or Mac emulator. The DDLab code, written in C, can also be recompiled following instructions in `readme` files and `Makefiles` provided.



$k = 3$ $k = 4$ $k = 5$

Fig. 15. For $k = 3$, 4, and 5, 20 overlaid entropy-density plots for just a few pulsing cycles each, and linking successive dots giving a time-history.

Read #4.1 (Quick Start Examples) for details about user input and control. Briefly, flashing cursor prompts are presented in turn; respond with input followed by **Return** to step forward, **q** to backtrack or interrupt. **Return** without input selects a default.

10.1. Basic steps to set up a CAP model

For CAP pulsing experiments, we will set up a 100 × 100 2d network where $v = 3$, and $k = 7$ (for example) and totalistic rules (#4.9.3).

(1) At the first prompt, enter **t** for TFO-mode (Totalistic Forward Only, #6.2.1).

(2) At the next **Value range** prompt enter **3** (#7.1).

(3) Enter **return** until a top-right **WIRING** prompt window appears (#11.1). Enter **2x** for hexagonal 2d (**2s** for square/orthogonal).

(4) At the next top-right prompt to set the 2d size (#11.6.1) enter **100** for both **i** and **j**.

(5) At the next top-right prompt, **Neighbourhood size k:** (#11.7) enter **7**.

(6) At the next top-right **2d network . . . wiring** prompt (#17.1), enter **2** to display a 2d wiring graphic bottom left, showing CA neighbourhoods, together with a top-right reminder (#17.4).

(7) Now set random wiring: enter **b** for a 2d block (#17.7.5), then **a** to outline the "block" as the whole network (#17.4 reappears), then **r** for random, which shows links for the highlighted cell as in Fig. 3 (a). Click other locations to show the wiring for other cells.

(8) Enter **return** until the top-right prompt **revise from:** (#31.1), enter **e**, then at the **entropy/density:** prompt (#31.5), enter **e** again.

(9) At the next top-right prompt, enter **d** to skip further special options and start the 2d space-time patterns iterating in the top-left of the screen for a random rule, with the input-entropy plot (averaged for a moving window of 10 time-steps) and the input-histogram alongside. The rule was set at random so oscillations are unlikely. To revise any of the above, backtrack with **q**.

10.2. Space-time patterns on-the-fly options

While 2d space-time patterns are running, on-the-fly key-press options may be activated/toggled (#32.3) — a reminder appears on the right of the screen. To pause (or backtrack) at any time key-press **q**, which gives a top-right prompt window with further options (#32.16). For the CAP model experiments, the following options are the most relevant.

(1) For a random rule from a glider collections[5], key-press **g** (#32.6.1).

(2) For a specific rule (thereafter consecutively), enter **q** to backtrack to the pause prompt (#32.16).

— enter **G** for a top-right prompt showing the number of rules in the collection (#31.2.9);

— enter **1**, or any valid number, to select the rule (not yet activated);

— once space-time patterns resume, key-press **g** to activate the rule;

— For the next rule enter **g**, eventually cycling back to rule 1.

(3) Waveform output is most pronounced when set to each single time-step, not to the default 10 time-step moving average. To change this, key-press **G** and at a top-right prompt (#32.12.7) enter enter **1**.

(4) Important on-the-fly key-presses:

— toggle with **7** between random-wiring and regular CA;

— try the 3-way toggle **j** for input-histogram values plotted together with the input-entropy plot;

— key-press **4** for a new random initial state. This is also required if the dynamics stops — reaches and attractor [4];

[5] In the $k = 6$ and $k = 7$ collections, not all rules are glider rules, so not all will pulse.

- — key-press **c** and **e** to contract/expand space-time patterns. Contract to allow more room for the entropy-density scatter plot below.

(5) Toggle showing the entropy-density scatter plot with **u** (#32.12.6). While this is running,

- — key-press **,** (comma) to toggle linking successive dots as in Fig. 5 (c);
- — key-press **?** (question mark) to toggle re-randomising at each time-step, which slows down iteration;
- — key-press **“** (inverted comma) to toggle showing running data of the wave-length (wl) and wave-height (wh) in the terminal as in Section 7. To save the data refer to #32.12.6.2.

(6) To change to value-density instead of entropy, toggle with **s**. The input-histogram (and entropy-density scatter plot) will stop updating.

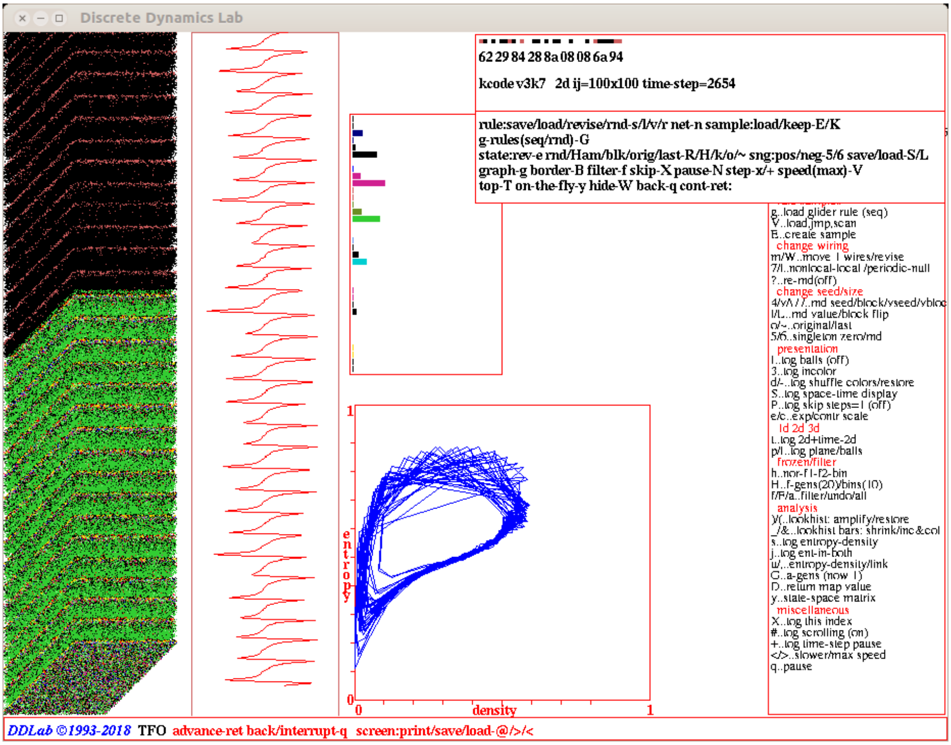
- — key-press **;** (semi-colon) to toggle showing the density return-map (#32.12.7);
- — key-press **,** (comma) to toggle linking successive dots as in Fig. 6 (c);
- — To save the density return-map data refer to #32.12.7.2.

(7) Other useful on-the-fly key-presses:

- — key-press **3** to toggle space-time patterns coloured by value or by neighbourhood/input-histogram colours;
- — key-press **t** to toggle between a 2d movie and 2d vertical space-time patterns as in Fig. 16, then # to toggle upward scrolling;
- — when in normal 2d, key-press # to toggle space-time patterns scrolling diagonally;
- — key-press < to slow down iteration, > to revert to normal speed.

(8) Key-press **q** to pause at any time for a top-right window providing options described in #32.16, including,

- — enter net-**n** to re-set the random wiring (#17.7) as in Sect. 10.1 (7), or with biases as in Fig. 3;
- — key-press **q** as necessary to backtrack up the DDLab prompt sequence;
- — key-press **d** to save wave-length data, or density return-map data (#32.12.6.2 and #32.12.7.2).



```
62 29 84 28 8a 08 08 6a 94

kcode v3k7   2d ij=100x100 time-step=2654
```

```
rule:save/load/revise/rnd-s/l/v/r net-n sample:load/keep-E/K
g-rules(seq/rnd)-G
state:rev-e rnd/Ham/blk/orig/last-R/H/k/o/~ sng:pos/neg-5/6 save/load-S/L
graph-g border-B filter-f skip-X pause-N step-x/+ speed(max)-V
top-T on-the-fly-y hide-W back-q cont-ret:
```

```
DDLab ©1993-2018 TFO advance-ret back/interrupt-q screen:print/save/load-@/>/<
```

Fig. 16. The DDLab screen showing 2d space-time patterns, 100 × 100, scrolling vertically upwards on the left. The present time-step is at the bottom. The dark time-steps at the top are coloured by value, and below by neighbourhood colours. To the right of space-time patterns is the entropy plot and the input-histogram, below centre the entropy-density plot with linked dots. Interrupt prompts are shown top-right, and on-the-fly reminder on the right. The rule is v3k7 $g3$ (hex)622984288a08086a94.

10.3. One cell at a time sequential updating

To run an entropy-density plot for sequential updating one cell at a time at random positions [1], as for the $k = 7$ rules in Fig. 14, amend steps in Section 10.1 as follows;

- — After step (7) enter **return** until the top-right prompt **revise from:** (#31.1), enter **u** for a top-right updating window (#31.4).
- — Enter **p** for a top-right partial order updating window (#31.4.3). For both **min:** and **max:**, enter **1**.
- — At the **entropy/density:** prompt (#31.5), enter **e** as in step (8).

Follow further steps as listed, then toggle showing the entropy-density scatter plot with **u** as in step (5) in Section 10.2. You will see single-cell updates in the 2d pattern, and a very slow trace of the plot. For other "asynchronous and noisy updating" options refer to (#31.4). There can be any combination of these settings, which can be toggled on-the-fly (#32.4).

11. Issues to explain the CAP model

Work explaining the CAP model is in progress. The emergence of gliders in CA cannot be predicted directly from a rule-table, so in this sense, the mechanisms are unresolved — they can only be observed by experiment, but must entail feedbacks driving a glider's head and eroding its tail. A general theory to resolve this question would shed light on the underlying principles of self-organisation. Pulsing when the wiring is randomised must utilise the same feedbacks, but distributed throughout the network instead of localised in a regular neighbourhood to create and move a glider.

Future work should also address the following issues:

- Because the pulsing waveform (shape/phase, wave-length, wave-height) is observed to be diverse, how does the type of glider dynamics relate to the waveform?
- Study how the pulsing waveform breaks down as the random wiring reach is reduced — is there a phase transition?
- There is a very high probability that gliders imply pulsing, but the few observed exceptions should be examined, gliders/no pulsing, and pulsing/no gliders.
- Study how the sequential updating of one cell at a time at random positions [1] can results in pulsing.

12. The need for a bio-oscillation model

In a world where much biology is produced by reproducing patters, produces reproducing patterns, or recognises these patterns, there has historically been focus on the chemistry and physics of thermodynamic equilibrium more so than on the bio-physics of collective oscillatory phenomena. Oscillations can be found in all forms of life [12], but we have focused on mammalian biology, and aspects of human physiology where oscillations play a crucial role [1].

Although differential equation models of oscillations in single cells have been proposed, such as the Hodgkins–Huxley equations, negative feedback with a time delay, or coupled negative and positive feedback [13], currently there is no satisfactory theory to explain essential oscillations in whole organs, for example, the heart beat, uterine contractions in childbirth, and various rhythmic behaviours controlled by the central pattern generators of the central nervous system, such as breathing and locomotion [1].

We are left searching for pacemaker neurones, pondering how signalling in biofilms can occur faster than diffusion, how synchronisation can occur over considerable distance and how biology is so robust with such inbuilt redundancy.

We propose that clusters of excitable tissue are able to oscillate according to their appropriate waveform because non-localised connectivity [14, 15] is subjected to a specific rule of communication, the bio-rule, analogous to a glider rule. The bio-rule is based on three (or more) cellular states: Firing, Refractory, and Ready-to-fire, generated by chemical signalling [16], action potentials, calcium and sodium ion channels and concentrations. This model is favoured by evolution because a variety of synchronized waveforms can arise from different bio-rules. Furthermore, a given waveform is independent of the exact connection network, is noise tolerant, can be turned off and on by altering the reach of non-local connectivity, and is robust to noise, cell loss, and functional reserve.

The CAP model carries the ability and benefits of modelling multiple cells simultaneously forming a platform to probe the relationship between network connectivity at one level and collective behaviour at another. This summary of the more detailed reasoning presented in [1] suggests that the CAP model is applicable as a conceptual model for bio-oscillations and can provide a basis for further development of the ideas.

13. Summary

We have presented further evidence and results for this surprising phenomenon of spontaneous, sustained and robust rhythmic oscillations, pulsing dynamics, when random wiring is applied to a 2d “glider” rule running in a 3-value totalistic CA.

We have reiterated the potential of the CAP model to provide a much needed decentralised model for bio-oscillations in nature, specifically in the case of mammalian excitable tissue.

We have defined the system’s architecture, and identified the behaviour for both glider dynamics and pulsing, which are intimately related, and noted the issues that require explanation. A guide to the relevant functions in the software DDLab to repeat and extend pulsing experiments is provided. However, the underlying mechanisms remain unresolved and are open to further study and research.

Experiments and figures were made with DDLab (http://www.ddlab.org/) — where the rules and methods are available, so repeatable [5]. Thanks to Inman Harvey for conversations regarding asynchronous updating, to Terry Bossomaier for exchanges regarding phase transitions, and to Paul Burt and Muayad Alasady for comments regarding bio-oscillations.

REFERENCES

[1] A. Wuensche, E.C. Coxon, arXiv:1806.06416 [nlin.CG], to appear in *J. Cell. Autom.*

[2] S.A. Kauffman, *J. Theor. Biol.* **22**, 437 (1969).

[3] B. Derrida, G. Weisbuch, *J. Physique* **47**, 1297 (1986).

[4] A. Wuensche, M. Lesser, *The global Dynamics of Cellular Automata*, Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, Reading, MA, 1992.

[5] A. Wuensche, *Exploring Discrete Dynamics — Second Edition*, Luniver Press, 2016, http://www.ddlab.org/download/dd_manual_2018/

[6] A. Wuensche, A. Adamatzky, *Int. J. Mod. Phys. C* **17**, 1009 (2006).

[7] A. Adamatzky, A. Wuensche, B. De Lacy Costello, *Chaos Solitons Fract.* **27**, 287 (2006).

[8] A. Adamatzky, A. Wuensche, *Complex Syst.* **16**, 277 (2006).

[9] A. Wuensche, *Complexity* **4**, 47 (1999).

[10] A. Wuensche, *Int. J. Unconv. Comput.* **1**, 375 (2005).

[11] A. Wuensche, Discrete Dynamics Lab (DDLab), 1993–2018, http://www.ddlab.org/

[12] Y. Cao, A. Lopatkin, L. You, *Nat. Struc. Mol. Biol.* **23**, 1030 (2016).

[13] T.Y.C. Tsai *et al.*, *Science* **321**, 126 (2008).

[14] A. Guo *et al.*, *Cardiovascular Res.* **98**, 204 (2013).

[15] R. Singh *et al.*, *Phys. Rev. Lett.* **108**, 068102 (2012).

[16] H. Cheng, W.J. Lederer, *Physiol. Rev.* **88**, 1491 (2008).