

Majority Gates and Circular Computation in Slime Mould

Genaro J. Martínez^{1,2}, Andrew Adamatzky¹, Richard Mayne¹, Fangyue Chen³, Qinbin He⁴

¹ Unconventional Computing Centre, University of the West of England, BS16 1QY Bristol, United Kingdom

² Escuela Superior de Cómputo, Instituto Politécnico Nacional, México

³ School of Sciences, Hangzhou Dianzi University, Hangzhou, Zhejiang 310018, P. R. China

⁴ Department of Mathematics, Taizhou University, Linhai, Zhejiang 317000, P. R. China

Abstract

Slime mould has been proven to be a fruitful living substrate for implementing a wide range of computing circuits from computational geometry to collision-based logical circuits to robot control. It is apparent, however, that constructing working real-time universal processors from the slime mould is non-trivial task. We explore here similarities between development of slime mould protoplasmic tubes, transportation of cytoplasm inside the tubes and dynamics of propagating patterns in cellular automata. Based on these analogies we will propose computing devices realisable in the living slime mould.

Keywords: unconventional computing, slime mould, mobile localizations, competing patterns, logic gates, tubes, cellular automata.

Preliminaries

Slime mould *Physarum polycephalum* is a syncytial single celled eukaryotic organism that is macroscopic in scale and possesses many thousands of nuclei. Both slime mould behaviour patterns and intracellular processes can be interpreted as expressions of unconventional computation. Achieving motility by the contraction of muscle proteins which drive cytoplasmic flows (cytoplasmic streaming), *P. polycephalum* may navigate between spatially distributed nutrient sources, forming an efficient, interconnected networks of tubular structures (protoplasmic ‘tubes’) in its wake which comprise the body of the cell.

Recently, a number of relevant and interesting experiments were performed to simulate computations and other slime mould phenomena (Adamatzky, 2010), including logic gates (Adamatzky et al., 2010), spacial logical gates (Schumann and Adamatzky, 2011), optically-coupled logic gates (Mayne and Adamatzky, 2015), hybridised slime mould (Mayne and Adamatzky, 2013), microfluidic logic gates (Adamatzky and Schubert, 2014), oscillators (Adamatzky, 2014), and to a number of diverse *Physarum* machines (Adamatzky, 2009). Several of them were implemented with specific initial conditions, where chemoattractants were used to control the slime mould propagation and modify topology of its protoplasmic networks.

We will discuss draw analogies between cytoplasm transportation inside the protoplasmic tubes, growth of the tubes and patterns growing in one- and two-dimensional cellular automata and speculate on what kind of computational devices can be mapped from the field of cellular automaton computing to slime mould computing devices.

Cytoplasm, protoplasmic tubes, and logic gates

A number of different logic gates derivations were reported with slime mould including conventional logic gates (Tsuda et al., 2004), ballistic logic gates (Adamatzky, 2010), artificial adders (Jones and Adamatzky, 2010), spatial logic gates (Schumann and Adamatzky, 2011), frequency logic gates (Adamatzky et al., 2014), microfluidic logical gates (Adamatzky and Schubert, 2014) and optically-coupled logic gates (Mayne and Adamatzky, 2015). Typically, the stream of information in these propagations are controlled on some T, Y, X, C, K, H-shaped junctions, or some variant of junctions (some samples are illustrated in Fig. 1). Following this principles, we can explore a kind of computation based in competing patterns that can be extrapolated to slime mould quickly.

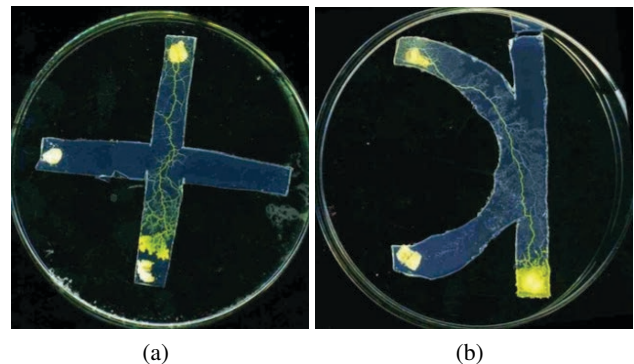


Figure 1: Photographs of exemplar slime mould computing landscapes implementations. (a) X junctions, (b) K junctions.

Cytoplasmic streaming and concurrent propagation of

plasmodial tubes along junctions are commonly used in designs of logic gates. In unstimulated, vegetative plasmodia, the organism will propagate towards a food source and cytoplasmic flows oscillate rhythmically through the hydrodynamic core of plasmodial tubes to distribute the absorbed nutrients. Where internalised substances are used to represent data, cytoplasmic streaming may be influenced experimentally by external stimuli (tactile, optical etc.) which will usually reduce local flow in a tube for several minutes. This principle has been used in the fabrication of some of the aforementioned Physarum logic devices. In existing designs of slime mould-based devices the topology of protoplasmic tubes is determined by configurations of attractants and repellents.

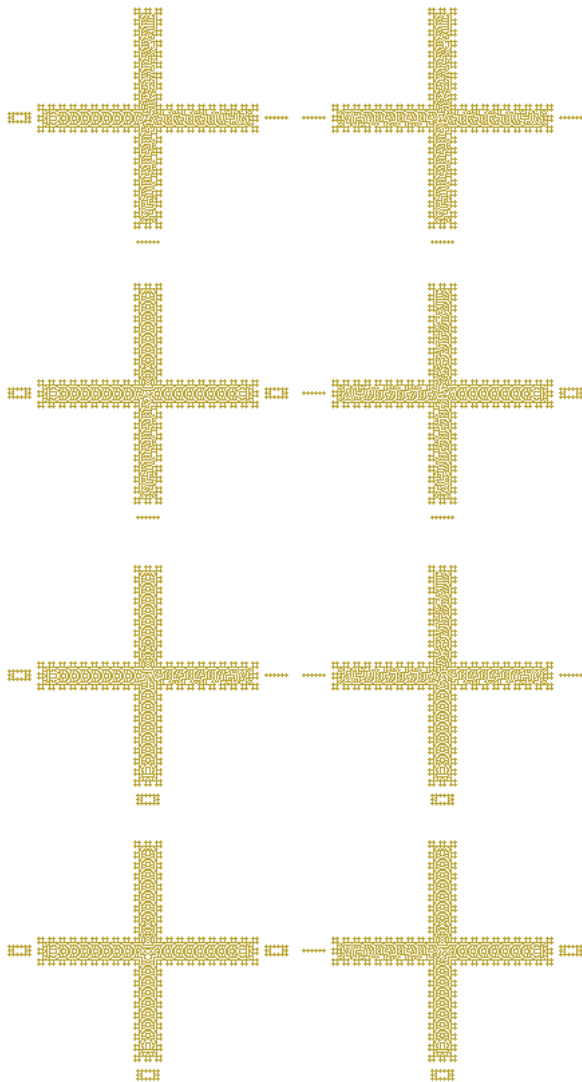


Figure 2: Idealised MAJORITY gate implemented in non-linear media (computing patterns) in artificial tubes (as networks in slime mould protoplasmic tubes).

Cytoplasmic flow is difficult to influence in any other way, however, as plasmodial behaviour is determined by myriad external and internal factors. In compensation for this, an opportunity to represent binary values can be realised by searching for symmetries in flow patterns through which simple logic gates by pattern propagation may be implemented (Martínez et al., 2008), (Martínez et al., 2010). The representation is simple in the first instance as cytoplasmic streaming must be controlled as a symmetric stream or asymmetric (zero or one). In this way, a number of logical gates can be implemented following a truth table. A salient limitation of this model is, however, that plasmodial tubes are fixed and cannot be removed easily. To solve this situation we can work with cascaded logic gates implemented by analogy with cellular automata, and although cellular automata dynamics seems simple from their own definition, they are powerful tools in simulating complex or chaotic systems. Cellular automata have been demonstrated to be as powerful as Turing machines and there are a number of cellular automata capable of universal computation (Adamatzky, 2002), (Hey, 1998), (Mitchell, 2001), (Martínez et al., 2013), (Wolfram, 1984).

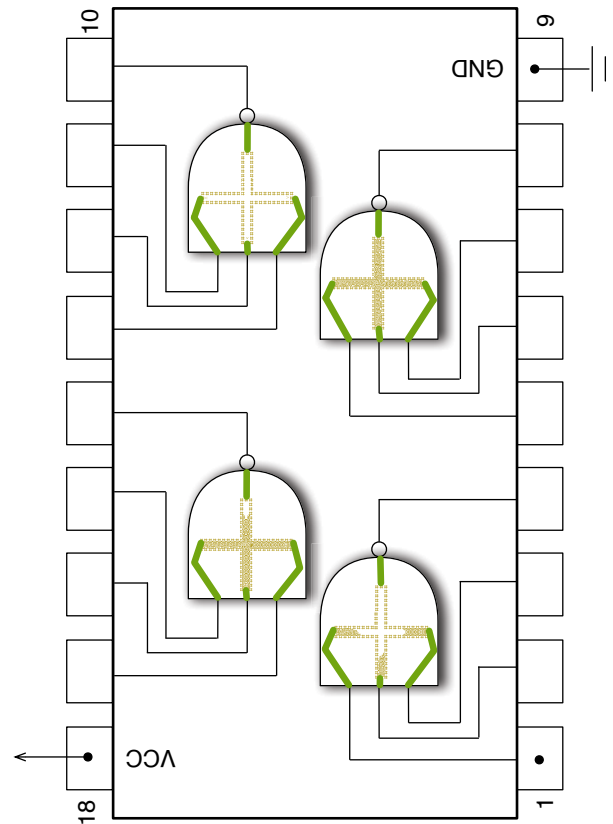


Figure 3: Device 5400/DM5400/DM7400 Quad 2-Input NAND Gates modified to works with MAJORITY gates based competing patterns.

Conventional AND, OR, NOT gates are relatively easy to implement in the Life-like cellular automata (Martínez et al., 2010). However, when implementing non-serial logical gates it is more convenient to work with MAJORITY gates, as they can be extrapolated to slime mould in a more realistic way (these kind of gates are used recurrently in quantum dots cellular automata, see (Gregory et al., 1999)). Figure 2 shows how a MAJORITY gate operates. In this case, these patterns will be encoded within the interior of protoplasmic tubes. We can utilise the scheme to design a modified chip related to 7400 chip from National semiconductor web site. Device 5400/DM5400/DM7400 Quad 2-Input NAND Gates http://www.icpdf.com/NSC_datasheet/DM5400_pdf_546245/, but with four MAJORITY and NOT gates instead of four NAND gates, working with three independent inputs per gate on 18 pins. This circuit is designed and illustrated in Figure 3. This way, each MAJORITY gate produces outputs depending of the 3-input values (eight possibilities Fig. 2). Additionally, the NOT gate can be implemented in circuits of competing patterns dynamics, as it was done in *hot ice* computers (Adamatzky, 2009).

Plasmodial tube and circular computation

By loading the plasmodium with a range of nanomaterials (metallic particles, fluorescent latex beads etc.) via vesicular endocytosis or microinjection, plasmodial bioelectrical activity may be altered and collisions between fluorescent particles may be observed, both of which may be used to represent information storage and processing (Mayne et al., 2011), (Mayne and Adamatzky, 2013). Figure 4 shows two snapshots of the particles travelling and colliding with stationary or mobile elements.

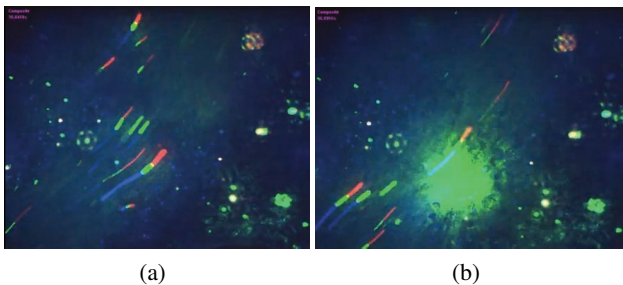


Figure 4: Confocal micrographs showing microscopic fluorescent particle dynamics in the *P. polycephalum* plasmodium (a) Particles traveling diagonally through the endoplasm of a plasmodial tube, and (b) shows the exact instant where an interaction is given. (Images captured from the video <https://www.youtube.com/watch?v=nYLmlF1V4sQ>)

Hypothetically, therefore, a number of plasmodial tubes carrying interacting exogenous particles may be used to design a digital circuit able to perform a computable func-

tion. Towards implementing such a system, two difficulties must be overcome: first, to control collisions between individual or groups of particles, and second, to reproduce and synchronise such a reactions periodically. These can be tackled by using our designs of *cellular automata collider* (Martínez et al., 2011) and approaches in *computing with rings* (Martínez et al., 2012).

An evolution space of one-dimensional cellular automata is able to produce particles, gliders, that emerge in complex patterns with a unique identity (period, speed, mass, volume, displacement). In this way, a number of particles are coded as regular expressions that can be manipulated to represent collisions and a number of nano component designs: computable devices, nano-structures, solitons, more complex particles, and beyond.

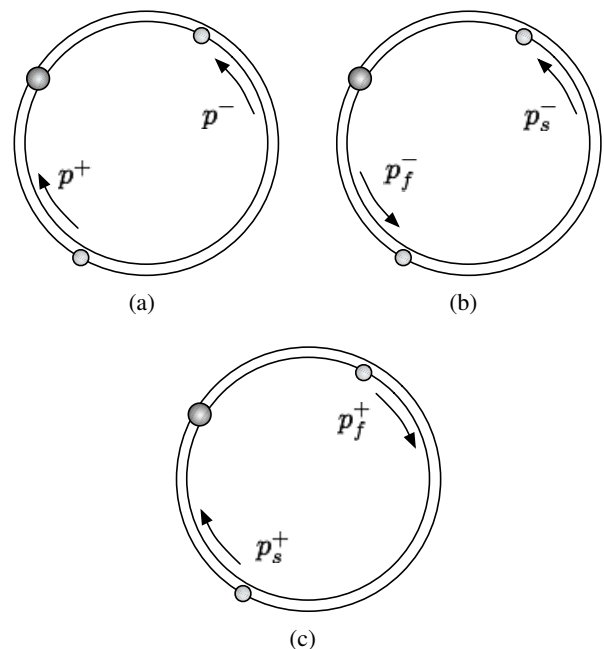


Figure 5: Representation of abstract particles in a one-dimensional cellular automata cyclotron.

Designs for computing these tubes follow a principle based on an unconventional computing representation paradigm (Mills, 2008). In this way, Fredkin and Toffoli have developed a concept of a general-purpose computation based on ballistic interactions between quanta of information that are represented by abstract particles (Fredkin and Toffoli, 2002). The Boolean states of logical variables are represented by balls or atoms, which preserve their identity when they collide with each other. Their ‘billiard-ball model’ of computation utilises underpinning mechanics of elastically colliding balls and mirrors reflecting the balls’ trajectories. Later, Margolus proposed a special class of cellular automata which implement the billiard-ball model. Margolus’ partitioned cellular automata exhibited computa-

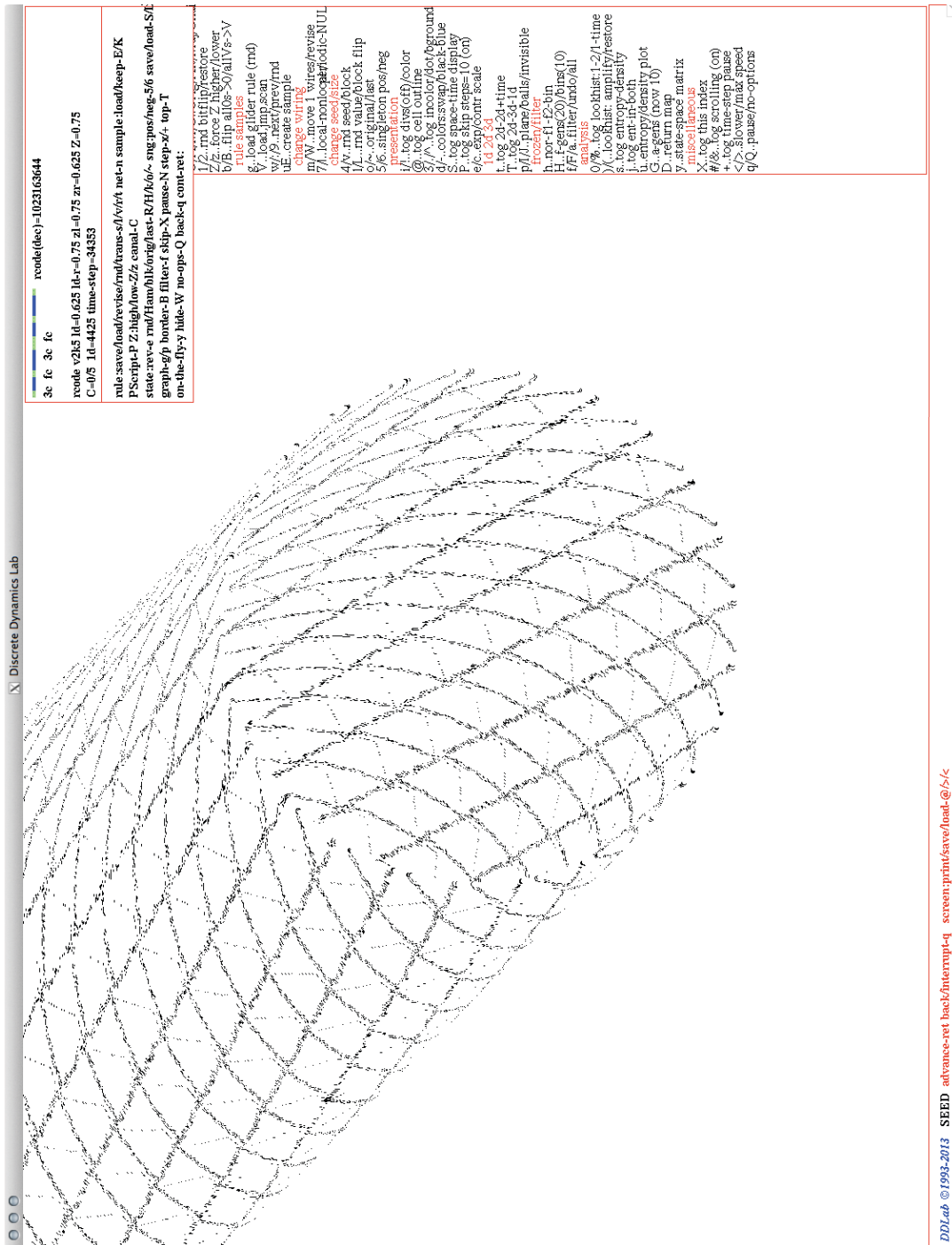


Figure 6: Cellular automata tube computing a simple oscillator simulating a structure such a carbon or graphene nanotube based particle collisions. Simulation done with DDLab (Dynamics Discrete Lab, <http://www.ddlab.org> (Wuensche, 2011).

tional universality because they simulated Fredkin gate via collision of soft spheres (Margolus, 2002).

Basic functions with two input arguments u and v can be expressed via collision between two localizations as shown in Fig. 6:

1. $f(u, v) = c$, fusion.
2. $f(u, v) = u + v$, interaction and subsequent change of state.
3. $f_i(u, v) \mapsto (u, v)$ identity, solitonic collision.
4. $f_r(u, v) \mapsto (v, u)$ reflection, elastic collision.

To map Toffoli's supercollider (Toffoli, 2002) onto a one-dimensional cellular automata we use the notion of an idealised particle $p \in \Sigma^+$ (without energy or potential energy). The particle p is represented by a binary string (regular expressions) of cell states (Martínez et al., 2011).

Figure 5 shows two typical scenarios where particles p_f and p_s travel in a cellular automata cyclotron. The first scenario (Fig. 5a) shows two particles travelling in opposite directions which then collide. Their collision site is shown by a dark circle in (Fig. 5a). The second scenario demonstrates a typical beam routing where a fast particle p_f eventually catches up with a slow particle p_s at a collision site, which is typically presented in particles emerging in protoplasmic tubes (Fig. 5b). If the particles collide like solitons (Jakubowski et al., 2001), then the faster particle p_f simply overtakes the slower particle p_s and continues its motion (Fig. 5c).

Following these basic principles, we can explore how microscopic fluorescent particles in *P. polycephalum* can be controlled and synchronised to manipulate computable devices in the large space of tubes/cyclotrons. We transduce the fluorescent particles as a set of strings and organise them in a set of valid reactions to convert a small database of these collisions to an algorithm.

The aim of such representation is to develop an automatic process to construct nano-assembly devices for unconventional computing, derived from a set of synchronised collisions between multiple particles. Such an automation is based on programming regular expressions and finite state machines on circular mechanism.

With regards to a complex one-dimensional cellular automaton supporting abstract particles emerging in its evolution space, we can codify a number of particles to synchronise multiple collisions and to simulate a basic oscillator. Figure 6 shows particles travelling in opposite directions: after of each collision between the particles two new particles emerge but they eventually collide and are transformed into the original two particles. These reactions are spatially synchronised and yield a basic oscillator projected on several copies. This pattern also can be seen as a nano-structure

constructed from collisions of mobile and stationary localizations in cellular automata and excitable media. This simulation starts with an initial condition of 4,425 cells codifying 25 particles evolving during 34,353 steps. The history of these collisions need 152,012,025 cells.

Cyclotrons and rings are powerful tools to implement sophisticated algorithms and complex codifications from other designers (Cook, 2004), (Wolfram, 2002), (Martínez et al., 2011). It is possible to design and codify complex equivalent Turing machines on these abstract colliders (following strong theories about circular Turing machines, circular Post machines, and cyclic tag systems) (Arbib, 1969), (Kudlek and Rogozhin, 2001), (Martínez et al., 2011).

Conclusion

Slime mould *P. polycephalum* is a powerful living computing substrate. When presented with data encoded in the configurations of attractants and repellents the slime mould works as an efficient specialised processor capable of approximating Voronoi diagrams, concave hull and shortest paths, and may solve many other computational problems. So far no general purpose computer chip has been developed with slime. This is due to difficulties in interfacing the protoplasmic tubes with conventional electronic components, due in part to the low conductivity of the protoplasmic tubes and instability of the tubes' topology. We have here provided a brief insight on how universal computing devices can be implemented with the slime mould's protoplasmic tubes. These analogies were drawn between competitive peristaltic contractions of the tube (cytoplasmic streaming) and a majority gate, where patterns growing inside the gate's channels compete for free space with each other, and cellular automata supercolliders, where gliders represent voltage solitons (Tuszyński et al., 2004) transported along actin filaments of protoplasmic tubes and the computation is implemented via collisions between the solitons. Hopefully, these abstract models will inspire us towards novel design of Physarum chips.

References

- Adamatzky, A. (ed.) (2002) *Collision-Based Computing*, Springer.
- Adamatzky, A. (2009) Hot ice computer, *Physics Letters A*, 374(2):264–271.
- Adamatzky, A. (2010) *Physarum Machines: Computers from Slime Mould*, World Scientific Series on Nonlinear Science, Series A.
- Adamatzky, A. (2010) Slime mould logical gates: exploring ballistic approach, arXiv:1005.2301 [nlin.PS].
- Adamatzky, A. (2014) Slime mould electronic oscillators, *Microelectronic Engineering Volume*, 124(25):58–65.

- Arbib, M. A. (1969) *Theories of Abstract Automata*, Prentice-Hall Series in Automatic Computation.
- Adamatzky, A. and Schubert, T. (2014) Slime mould microfluidic logical gates, *Materials Today*, 17(2):86–91.
- Adamatzky, A., Whiting, J., and Costello, B. L. (2014) Slime mould logic gates based on frequency changes of electrical potential oscillation, *Biosystems*, 124:21–25.
- Cook, M. (2004) Universality in Elementary Cellular Automata, *Complex Systems*, 15(1):1–40.
- Fredkin, E. and Toffoli, T. (2002) Design Principles for Achieving High-Performance Submicron Digital Technologies, In: *Collision-Based Computing*, A. Adamatzky (Ed.), Springer, chapter 2, 27–46.
- Hey, A. J. G. (1998) *Feynman and computation: exploring the limits of computers*, Perseus Books.
- Jones, J. and Adamatzky, A. (2010) Towards Physarum binary adders, *BioSystems*, 101:51–58.
- Jakubowski, M. H., Steiglitz, K., and Squier, R. (2001) Computing with Solitons: A Review and Prospectus, *Multiple-Valued Logic*, 6(5-6). In: *Collision-Based Computing*, A. Adamatzky (Ed.), Springer, chapter 10, 277–298.
- Kudlek, M. and Rogozhin, Y. (2001) New Small Universal Post Machine, *Lecture Notes in Computer Science*, 2138:217–227.
- Mayne, R., Patton, D., de Lacy Costello, B., Adamatzky, A. and Patton, R. (2011) On the internalisation, intraplasmoidal carriage and excretion of metallic nanoparticles in the slime mould, *Physarum polycephalum*, *International Journal of Nanotechnology and Molecular Computation*, 3(3):1–14.
- Mayne, R. and Adamatzky, A. (2013) Towards Hybrid Nanostructure-Slime Mould Devices, *Nano LIFE*, 4:1450007.
- Mayne, R. and Adamatzky, A. (2015) Slime mould foraging behaviour as optically-coupled logical operations, *International Journal of General Systems* DOI: 10.1080/03081079.2014.997528.
- Martínez, G. J., Adamatzky, A., and Costello, B. L. (2008) On logical gates in precipitating medium: cellular automaton model, *Physics Letters A*, 1(48):1–5.
- Martínez, G. J., Adamatzky, A., Morita, K., and Margenstern, M. (2010) Computation with competing patterns in Life-like automaton, In: *Game of Life Automata*, A. Adamatzky (ed.), Springer, chapter 27, pages 547–572.
- Martínez, G. J., Adamatzky, A., and McIntosh, H. (2012) Computing on rings In: *A computable Universe: Understanding and Exploring Nature as Computation*, H. Zenil (ed.), World Scientific Press, chapter 14, pages 283–302.
- Margolus, N. H. (1984) Physics-like models of computation, *Physica D*, 10(1-2):81–95.
- Margolus, N. H. (1998) Crystalline Computation, In: *Feynman and computation: exploring the limits of computers*, A. J. G. Hey (Ed.), Perseus Books, 267–305.
- Margolus, N. (2002) Universal Cellular Automata Based on the Collisions of Soft Spheres, In: *Collision-Based Computing*, A. Adamatzky (ed.), Springer, chapter 5, pages 231–260.
- Martínez, G. J., Adamatzky, A., Stephens, C. R., and Hoeflich, A. F. (2011) Cellular automaton supercolliders, *International Journal of Modern Physics C*, 22(4):419–439.
- McIntosh, H. V. (2009) *One Dimensional Cellular Automata*, Luniver Press, UK.
- Mills, J. W. (2008) The Nature of the Extended Analog Computer, *Physica D*, 237(9):1235–1256.
- Minsky, M. (1967) *Computation: Finite and Infinite Machines*, Prentice Hall.
- Mitchell, M. (2001) Life and evolution in computers, *History and Philosophy of the Life Sciences*, 23:361–383.
- Martínez, G. J., Morita, K., Adamatzky, A., and Margenstern, M. (2010) Majority adder implementation by competing patterns in Life-like rule B2/S2345, *Lecture Notes in Computer Science* 6079:93–104.
- Martínez, G. J., McIntosh, H. V., Seck-Tuoh-Mora, J. C., and Vergara, S. V. C. (2011) Reproducing the cyclic tag system developed by Matthew Cook with Rule 110 using the phases f_{1-1} ,” *Journal of Cellular Automata*, 6(2-3):121–161.
- Martínez, G. J., Seck-Tuoh-Mora, J. C., and Zenil, H. (2013) Computation and Universality: Class IV versus Class III Cellular Automata, *Journal of Cellular Automata*, 7(5-6):393–430.
- Margolus, N., Toffoli, T., and Vichniac, G. (1986) Cellular-Automata Supercomputers for Fluid Dynamics Modeling, *Physical Review Letters*, 56(16):1694–1696.
- Schumann, A. and Adamatzky, A. (2011) Physarum Spatial Logic, *New Math. and Nat. Computation*, 7(3):483–498.

- Gregory, L. S., Orlov, A. O., Amlani, I., Bernstein, G. H., Lent, C. S., Merz, J. L., and Porod, W. (1999) Quantum-Dot Cellular Automata: Line and Majority Logic Gate, *Japanese Journal of Applied Physics*, 38:7227–7229.
- Tsuda, S., Aono, M., and Gunji, Y-P (2004) Robust and emergent Physarum logical-computing, *BioSystems*, 73:45–55.
- Toffoli, T. (1998) Non-Conventional Computers, *Encyclopedia of Electrical and Electronics Engineering* (John Webster Ed.), 14:455–471, Wiley & Sons.
- Toffoli, T. (2002) Symbol Super Colliders, In: *Collision-Based Computing*, A. Adamatzky (Ed.), Springer, chapter 1, 1–22.
- Tuszyński, J. A., Portet, S., Dixon, J. M., Luxford, C., and Cantiello, H.F. (2004) Ionic wave propagation along actin filaments, *Biophysical journal*, 86:1890–1903.
- Wolfram, S. (1984) Universality and complexity in cellular automata, *Physica D*, 10:1–35.
- Wolfram, S. (2002) *A New Kind of Science*, Wolfram Media, Inc., Champaign, Illinois.
- Wuensche, A. (2011) *Exploring Discrete Dynamics* Luniver Press, UK.