Studia Humana
Volume 1:1 (2012), pp. 96 — 101.

INTERVIEW
IS EVERYTHING A COMPUTATION?

Andrew Adamatzky is Professor in Unconventional Computing in the Department of Computer Science, amir of the Unconventional Computing Centre, and a member of Bristol Robotics Lab. He does research in reaction-diffusion computing, cellular automata, physarum computing, massive parallel computation, applied mathematics, collective intelligence and robotics. He is one of the founders of unconventional computing thinking in natural sciences.

Andrew Schumann: What is unconventional computing? How does it differ from other approaches to computation?

Andy Adamatzky: Usually, in answering this question I could not resist quoting Tomasso Toffoli: "... a computing scheme that today is viewed as unconventional may well be so because its time hasn't come yet — or is already gone." This means that everything flows and nothing stays the same, e.g. at the time of analogue computers digital ones were considered unconventional, but nowadays they top the charts of modern unconventional computing devices.

The simplest explanation what the conventional computer is may be as follows. A conventional processor converts all the inputs of keyboard into binary numbers (0s and 1s in the form of electrical pulses) and from then on its function involves the movement and transformation of these pulses in simple electrical circuits. Its advantages lie in the fact that it has millions of such circuits operating at high speed and can thus 'compute' outputs very quickly.

However, the conventional processor has rigid limitations. Obviously, if the basic technology behind this conventional computation had endless development potential then any limitations might be easier to ignore. Nevertheless, there is a consensus that current methods will indeed reach a threshold and this has led to an explosion in research into unconventional methods of computation.

The main limitations are in that conventional processors compute in a serial manner whereas biological and natural information processing seems to be predominantly via parallel mechanisms. Conventional processors are hard wired while unconventional ones are soft-, chemical- and molecular-based devices. Conventional computers are fragile, in a sense that damaging one component will usually halt the work of the whole machine, and unconventional ones are 'self-healing', re-constructible, due to the behavior of the physical matter they are built of.

Recently, unconventional computing is a huge area of joint researches of computer scientists, biologists, physicists, etc. The subjects commonly addressed in unconventional computing research are as follows now:

- *cellular automata* (the way of designing models, topological conformity to natural spatially extended systems and huge potential to exhibit all types of complex behavior with simple local transition rules);
- *biological and molecular computing* (conformation-based computing, DNA computing, information processing in micro-tubules, molecular memory, biochemical computing, artificial chemistry, etc.);

- *chemistry-based computing* (amorphous computing, implementation of logical functions, image processing and pattern recognition in reaction-diffusion chemical systems and networks of chemical reactors);
- *hybrid and non-silicon computation* (plastic computers, organic semiconducting devices, neuronal tissue-silicon hybrid processors);
- *logics of unconventional computing* (logical systems derived from space-time behavior of natural systems, non-classical logics, logical reasoning in physical, chemical and biological systems);
- *physics-based computation* (analogue computation, quantum computing, collision-based computing with solitons);
- *stigmergic and population-based computing* (optimization in cellular cultures, computing in societies of social insects, ecological computing);
- *smart actuators* (molecular motors and machines with computational abilities, intelligent arrays of actuators, molecular actuators, coupling unconventional computing devices with arrays of molecular or smart-polymer actuators).

Thus, the research in unconventional, or nature-inspired, computing aims to uncover novel principles of efficient information processing and computation in physical, chemical and biological systems, to develop novel non-standard algorithms and computing architectures, and also to implement conventional algorithms in non-silicon, or wet, substrates.

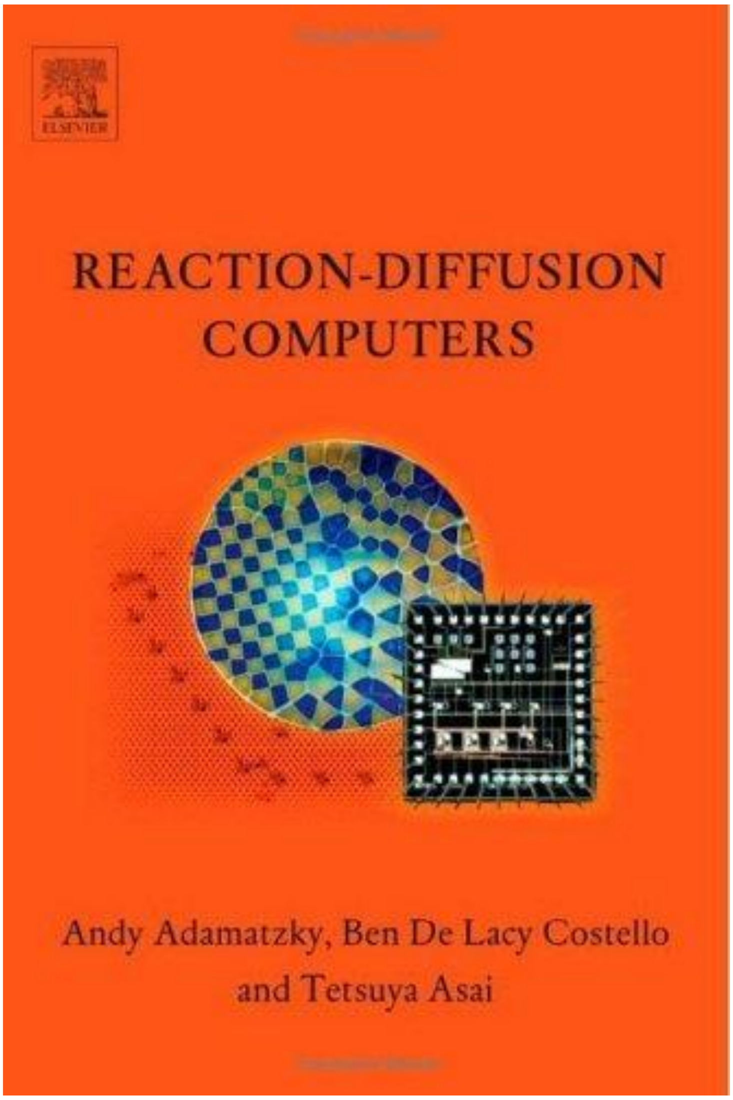
To sum up, in a word it is difficult to draw a clear borderline between ‘conventional’ and ‘unconventional’ computing. As I said, analog computation was conventional half-a-century ago, then digital computing came to power, and now the analog computation is ‘unconventional.’ Memristors is another good example here. Memristors (resistors with memory proposed by Leon Chua in 1973) were purely theoretical curiosities years ago. When fabricated in HP Labs in 2008, they became ‘unconventional’ for a year or two. Now hundreds of companies are producing memristors, and thousands of papers are published on memristors. Are memristors ‘unconventional’ or ‘conventional’ recently? For some guys ‘unconventional’ means quantum computing, for others reaction-diffusion chemical computing, for others ant-based algorithms and so on.

A.Sch.: What is provided by combining physics, chemistry and biology within the theory of computation?

A.A.: Novel principle of information processing, e.g. wave-based computation and collision-based computing, and novel substrates for computation. But, most importantly, years of fun and entrainment by playing with unknowable and exploring the ways the Mother Nature thinks.

A.Sch.: What is reaction-diffusion computing that was proposed by you? Which features distinguishing it from other unconventional computing approaches does it have?

A.A.: A reaction-diffusion processor is a real chemical medium, usually composed of a thin layer of solution or gel containing chemical reagents, which in its space-time dynamics transforms data to results in a sensible and programmable way. Data, to be processed, can be represented by the concentration of certain reagents and spatial structures, e.g. diffusive or excitation waves, spread from these initial data points. The spreading patterns interact to produce either stationary structures, e.g. a precipitate concentration profile, or dissipative structures, e.g. oscillating patterns. The final state, or even just a particular spatial state of the whole medium, represents a result of the reaction-diffusion computation. The spreading of waves is analogous to information transfer. And, the interaction of diffusive or phase waves realizes the computation.



An important attribute of this mode of computation is that there is an absence of a rigid hardware-like structure. Essentially, the ‘liquid’ processor has an ‘amorphous’ structure which may be considered as a layer of micro-volume reaction-diffusion chemical processors capable of massive parallelism. Characteristic advantages of reaction-diffusion processors include parallel input of data (usually, via the spatial distribution of the reactant concentrations), massively parallel information processing (typically, via spreading and interaction of either phase or diffusive waves) and parallel output of results of the computation (commonly, the results are represented by patterns of reactants or a colored precipitate that enables the use of optical reading devices).

These features together with the relative ease of laboratory experiments (most reactions occur at room temperature and do not require any specialist equipment), constructional simplicity of formal design (all reaction-diffusion systems are well simulated in two-dimensional cellular automata) and the pleasure of parallelism per se make reaction-diffusion chemical processors an invaluable tool for developing advanced unconventional parallel computing architectures.

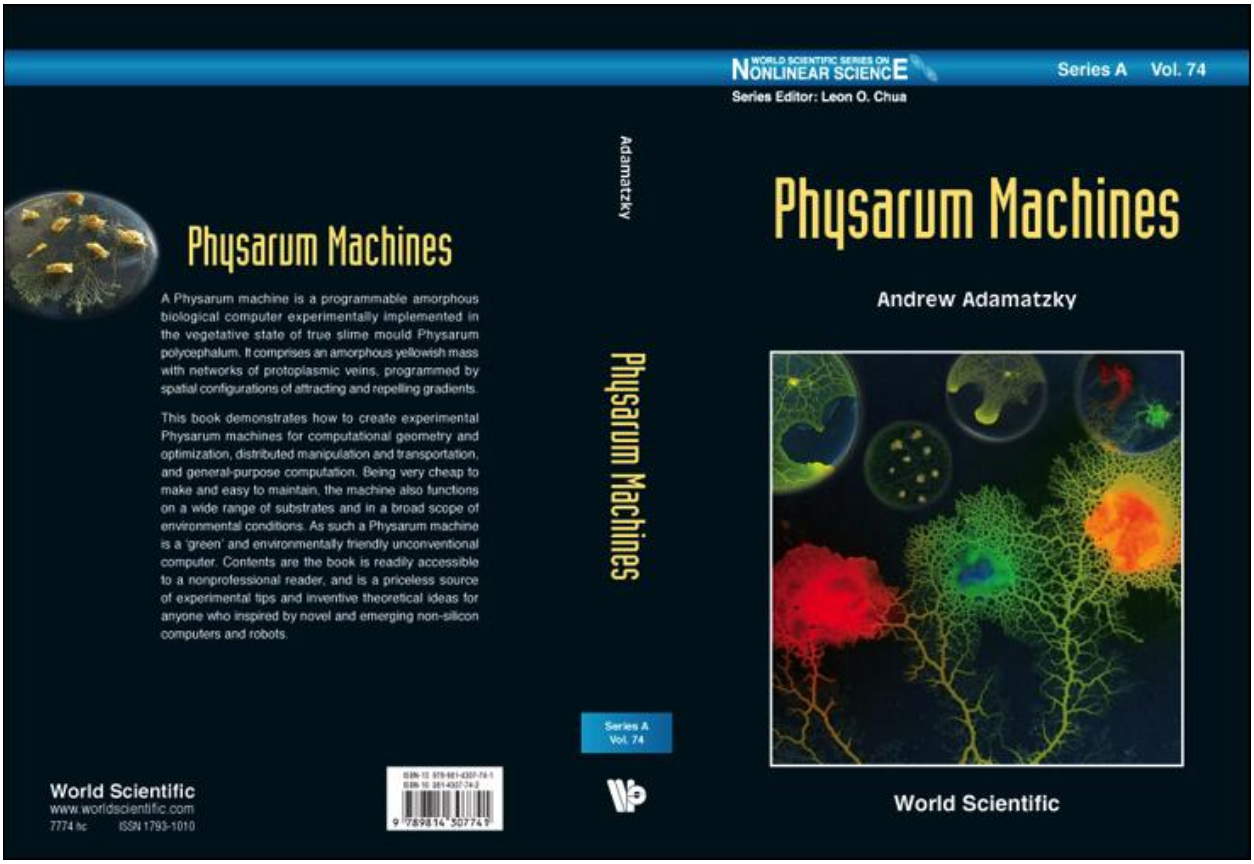
For more details you can see our book Adamatzky A., De Lacy Costello B, Asat T. *Reaction-Diffusion Computers* (Elseviers, 2005).

A.Sch.: One of your latter investigations concerns *Physarum polycephalum* computing. Why are you going to implement the computations on the medium of *Physarum*? Could we say that *Physarum polycephalum* is intelligent?

A.A.: Physarum polycephalum is a very user-friendly creature. It does not require any specialist laboratory equipment. Anyone can do experiments with *Physarum* at the office, kitchen and even bedroom. (I guess we discuss experiments in bedroom in some other interview.) So, yes, when in 2006 Soichiro Tsuda sent me samples of *Physarum*, I thought at first that it is total crap, but then I started to play and realized that it is a wonderful creature indeed, very responsive, adaptable and capable to solve numerous problems of computational geometry, mathematical logic and even navigate robots. Since then I have developed a concept and experimental laboratory implementations of *Physarum* machines.

What is a *Physarum* machine in a word? A *Physarum* machine is a programmable amorphous biological computer experimentally implemented in plasmodium of *Physarum polycephalum*. The point is that a *Physarum* machine on a nutrient-rich substrate behaves as an auto-wave in an excitable medium. On a non-nutrient substrate it propagates similarly to a wave fragment in a sub-excitable medium. A *Physarum* machine can be programmed by configurations of repelling (salt) and attracting (food) gradients, and localized reflectors (illuminated obstacles).

What *Physarum* machines can do first of all? It solves mazes. A *Physarum* machine represents a path from start to finish sites in a maze by its protoplasmic tube. It approximates a planar Voronoi diagram. Data planar points are mapped by pieces of plasmodia. Bisectors of the Voronoi diagram are composed of the substrate's loci not colonized by plasmodium. As a result, a *Physarum* machine computes a nearest-neighborhood graph, a spanning tree, a relative neighborhood graph and a Delaunay triangulation at various stages of its development. Nodes of a graph are represented by sources of nutrients, edges by protoplasmic tubes connecting the sources.



Physarum Machines

A Physarum machine is a programmable amorphous biological computer experimentally implemented in the vegetative state of true slime mould Physarum polycephalum. It comprises an amorphous yellowish mass with networks of protoplasmic veins, programmed by spatial configurations of attracting and repelling gradients.

This book demonstrates how to create experimental Physarum machines for computational geometry and optimization, distributed manipulation and transportation, and general-purpose computation. Being very cheap to make and easy to maintain, the machine also functions on a wide range of substrates and in a broad scope of environmental conditions. As such a Physarum machine is a 'green' and environmentally friendly unconventional computer. Contents are the book is readily accessible to a nonprofessional reader, and is a priceless source of experimental tips and inventive theoretical ideas for anyone who inspired by novel and emerging non-silicon computers and robots.

World Scientific
www.worldscientific.com
7774 hc ISSN 1793-1010

Adamatzky
Physarum Machines
Series A
Vol. 74

WORLD SCIENTIFIC SERIES ON NONLINEAR SCIENCE
Series A Vol. 74
Series Editor: Leon O. Chua

Physarum Machines

Andrew Adamatzky

World Scientific

Also, notice that a *Physarum* machine is a universal computer. The machine implements Boolean logic conjunction, disjunction and negation on a geometrically constrained substrate. The machine can also realize binary adders. Hence, we could claim that a *Physarum* machine is a programmable manipulator. The machine can push and pull objects floating on a water surface by expanding and contracting its protoplasmic tubes.

Thus, *Physarum* is a pretty university amorphous biological substrate. Moreover, a *Physarum* machine is a simple biological implementation of a Kolmogorov-Uspensky machine, i.e. a biological substrate for all general purpose computing devices. For details see my other book Adamatzky A. *Physarum machines* (World Scientific, 2010).

As concerns your last part of the question, *Physarum* is intelligent in fact as a drop water running down the window glass. *Physarum* just follows gradients of attractants and repellents. Nothing more.

A.Sch.: What else from physical or biological systems may be presented as process of computation? Everything? Why?

A.A.: Absolutely everything! The matter is that a computation is just our interpretation, our view. Computing potential of any biological, chemical or physical systems is determined only by phantasies of researchers who have built unconventional computers from these systems.

A.Sch.: Could we claim that unconventional computing is a novel paradigm in natural sciences that will absorb physics, chemistry, biology and other sciences in process of time?

A.A.: Unconventional computing is always in the flux. Some concepts become conventional new concepts and prototypes emerge. Unconventional computing is the art of interpretation, and we will always have plenty of phenomena to interpret. For example, a plasmodium, or vegetative state, of *Physarum polycephalum* behaves like a giant amoeba. It is possible to show that topology of the plasmodium's protoplasmic network optimizes the plasmodium's harvesting on distributed sources of nutrients and makes more efficient flow and transport of intra-cellular components. As a result, this dynamics could be interpreted as approximation of shortest path in a maze, computation of proximity graphs, Delaunay triangulation, construction of logical gates, robot control, implementation of storage-modification machines, approximation of Voronoi diagram, and a network of biochemical oscillators and so on. Hence, all depends on our interpretation and imagination. Just our fantasy holds.

References

1. Adamatzky A., Teuscher C., *From Utopian to Genuine Unconventional Computers*. Luniver Press, 2006.
2. Adamatzky A., De Lacy Costello B., Bull L., *Unconventional Computing 2007*. Luniver Press, 2007.
3. Antoniou I., Calude C.S., Dinneen M. J. (eds.), *Unconventional Models of Computation*, Springer-Verlag. London, 2000.
4. Calude C. S., Casti J. L. (eds.), "Unconventional Models of Computation", *Complexity*, Vol. 4, 1, 1998.
5. Calude C. S., Casti J., Dinneen M. (eds.)., *Unconventional Models of Computation*. Springer-Verlag, Singapore, 1998.
6. Calude C.S., Costa, J. F., Freund R., Oswald M., Rozenberg G. (eds.), *Proc. 7th International Conference Unconventional Computation*. Lecture Notes Comput. Sci. 5204, Springer, Heidelberg, 2008.
7. Calude C.S., Dinneen M. J., Peper F. (eds.), *Unconventional Models of Computation.* Lecture Notes Comput. Sci. 2509, Springer Verlag, Heidelberg, 2002.

8. Calude C.S., Dinneen M. J., Perez Jimene M. J., Paun G., Rozenberg G. (eds.). *Proc. 4th International Conference Unconventional Computation*. Lecture Notes Comput. Sci. 3699, Springer, Heidelberg, 2005.
9. Calude C.S., Hagiya M., Morita K., Rozenberg G. (eds.), *Proc. 9th International Conference Unconventional Computation*. Lecture Notes Comput. Sci. 6079, Springer, Heidelberg, 2010.
10. Teuscher C., Adamatzky A., *Unconventional Computing 2005: From Cellular Automata to Wetware*. Luniver Press, 2005.