PhysComp96: `cmlpow`
Extended abstract
Draft, 12 May 1996
`orponen:   orponen@math.jyu.fi`

Universal computation by finite two-dimensional coupled map lattices

Pekka Orponen[*]
orponen@math.jyu.fi
University of Jyväskylä, Dept. of Mathematics
P. O. Box 35, FIN–40351, Jyväskylä, Finland
Martin Matamala[†]
mmatamal@dim.uchile.cl
Universidad de Chile, Dept. de Ingeniería Matemática
Casilla 170/3 – Correo 3, Santiago, Chile.

1 Introduction

Coupled map lattices (CML's) are locally-coupled discrete-time, discrete-space, continuous-state dynamical systems. This cellular-automata-like discretization of PDE's was introduced in the early 1980's in studies of spatiotemporal chaos and other phenomena arising in reaction-diffusion processes. Since then CML's have been applied as models for problems in, e.g. solid-state physics, population biology, neurophysiology, and information processing (for an overview, see [6]).

The question of the general computational power of the CML model is rather interesting, both from the point of view of obtaining an understanding of the dynamical possibilities of the model, and with a view towards possible computational applications. Indeed in the concluding phrases of his survey article on CML's[6], Kaneko states: "A CML with chaotic behavior must have higher computational ability than a CA, since the former can create information. It will be important to clarify what a CML can do that the conventional computer cannot."

Kaneko's aim is actually set too high, as it is rather easy to see that given a finitely described initial state, and with effectively computable response functions, a finite CML can always be simulated on a Turing machine, *i.e.* a digital computer. In this paper we establish, however, that there are no other limitations to the computational power of the model: a universal Turing machine can be simulated on a two-dimensional CML with 74580 scalar lattice sites, and thus CML's are in principle capable of doing anything that a digital computer can do.

The local response functions of our CML model are simple piecewise-linear, unimodal maps similar to the tent map[2]. The sites in the lattice are interconnected in a symmetric von Neumann pattern (*i.e.*, each site is connected to its two vertical and two horizontal neighbors), with homogeneous and symmetric diffusion coefficients. However, in terms of the local response functions our lattice is strongly anisotropic: each site has its own response function. We leave it as an open question whether also completely isotropic lattices can be computationally universal.

2 Preliminaries

A **coupled map lattice** is a discrete-time dynamical system defined on a finite set of **sites** $k = 1, \ldots, N$. (We only consider finite systems in this paper.) We are mainly interested in systems where the instantaneous **states** of the sites are scalars $x_k(t) \in \mathcal{R}$, but for technical reasons we shall also consider systems where the states may be vectors $x_k(t) \in \mathcal{R}^2$.

Each site k has an associated **local response function** f_k, which maps site states to site states. A typical choice in dynamical system studies of CML's (see, e.g. [4, 5, 6]) is the logistic map $f(x) = ax(1-x)$, for some control parameter a. A piecewise-linear version of this is the tent map:

$$f(x) = \begin{cases} 2x, & \text{if } 0 \le x < 1/2, \\ 2(1-x), & \text{if } 1/2 \le x \le 1, \\ 0, & \text{otherwise.} \end{cases}$$

From a dynamical systems perspective[2], the important common property of both the logistic and the tent map is that they are **unimodal**, *i.e.* downwards concave with a unique maximum. For our purposes, we shall need to consider a broader class of unimodal maps of the form:

$$f(x) = \begin{cases} \alpha x + \beta, & \text{if } a \le x \le b, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

As a special case, a site may also have the identity response function—although even this may be made to formally conform to condition (1) by defining:

$$f(x) = \begin{cases} x, & \text{if } 0 \le x \le \mu, \\ 0, & \text{otherwise,} \end{cases}$$

*Most of this work was done during this author's visits to the Universidad de Chile, supported by the Fondo Nacional de Ciencia y Tecnologia grant Fondecyt 1940520, and to the Ecole Normale Supérieure de Lyon, supported by the ESPRIT Working Group NeuroCOLT 8556.

†Work done at U. de Chile, supported by Fondecyt grant 1950569, and at the ENS Lyon, supported by ECOS (Cooperation Scientifique Franco-Chilienne).

where μ is the maximum value of a state.

Most of the maps described by (1) are of course discontinuous. However, our constructions are in fact not sensitive to the behavior of the maps in small ranges beyond the discontinuities, and so all the maps considered could be made continuous by linear interpolation over small ranges of the form $[a-\epsilon, a]$ and $[b, b+\epsilon]$. Nevertheless, for simplicity we prefer to discuss our constructions in terms of the formally discontinuous maps of form (1).

The sites in a CML are interconnected according to some regular pattern, which we take to be the **von Neumann neighborhood** structure: in a one-dimensional lattice, each site k is connected to sites $k-1$ and $k+1$ (with a periodic boundary condition, so that the neighbors are actually determined mod N); in a two-dimensional system the sites are indexed by pairs $(i,j) \in N \times N$, and each site (i,j) has four neighbors $(i \pm 1, j)$ and $(i, j \pm 1)$ (mod N). In our constructions, we use "punctured" neighborhoods, *i.e.* a site is not a member of its own neighborhood.

The strength of the coupling between two neighboring sites k and l is indicated by a **diffusion coefficient** ϵ_{kl}. In our case all the diffusion coefficients will be equal and can be normalized to unity, so that the state update rule for a site k becomes:

$$x_k(t+1) = f_k(\sum_{l \in N_k} x_l(t)),$$

where f_k denotes the response function at site k, and N_k denotes the (1-D or 2-D) punctured von Neumann neighborhood of site k.

For technical reasons, we shall in our intermediate constructions discuss also lattices with asymmetric interconnections, where the diffusion coefficient from site k to site l is unity, and in the opposite direction zero. Extending the update rule to cover also this possibility is straightforward.

3 The construction

We start from the simulation of a universal Turing machine by a two-dimensional, piecewise-linear iterated function system described by Koiran *et al.* [7]. (For our purposes, this is the most useful one from a family of related constructions presented by, e.g. Asarin and Maler[1], Moore[9], and Siegelmann and Sontag[10].)

In the Koiran *et al.* construction (as in all the above mentioned simulations), the tape of a Turing machine M to be simulated is represented as two opposing stacks, whose contents are then encoded as two real numbers in the unit interval, using an appropriate Cantor encoding. (To be precise, also the state of the finite control of M is encoded on top of the stacks, and hence in the first few bits of their real-number encodings.) Thus, each configuration of M is represented as a point $(x_1, x_2) \in [0,1]^2$, and moreover in such a manner that the first few bits of x_1 and x_2 determine how the point is to be mapped by the iterated function $\sigma_M : [0,1]^2 \to [0,1]^2$ corresponding to the state transition function of M.

In fact, it turns out that the function σ_M is piecewise-linear, mapping each subsquare of the form $I_{qa_1a_2} \subseteq [0,1]^2$, determined by a state q of M and top-of-stack symbols a_1 and a_2, affinely into the set $\bigcup_{b_1b_2} I_{q'b_1b_2}$, representing the possible configurations of M after it has moved from state q to state q' and shifted material from one stack to another according to its single-step transition rule.

In order to simulate a Turing machine with n states and k tape symbols, one thus obtains a 2-D iterated function composed of nk^2 affine pieces. Applying this construction to Minsky's[8, pp. 277–280] 7-state, 4-symbol universal Turing machine yields a function with 112 components. (In fact, in order to extend the function from the subsquares of the form $I_{qa_1a_2}$ to all of $[0,1]^2$, one needs an additional $6nk^2$ interpolating triangles, bringing the total number of affine components up to 784 in the case of Minsky's machine.)

Summarizing, Koiran *et al.* thus obtain a computationally universal iterated function $\sigma : [0,1]^2 \to [0,1]^2$, composed out of 112 affine components

$$\sigma_k : I_k \to [0,1]^2, \qquad \sigma_k(x) = A_k x + \lambda_k,$$

where the I_k are disjoint subsquares of $[0,1]^2$, the A_k are real 2×2 -matrices, and the λ_k are real 2-vectors. (We are consistently ignoring the triangular components interpolating between the squares I_k.)

Our goal is to implement the map σ as a 2-D scalar-state CML, with symmetric diffusion coefficients between neighboring sites. As a first step, we observe that the map is easy to implement as a 1-D periodic CML chain with 113 asymmetrically interconnected, $\mathcal{R}^2$-valued sites (Fig. 1).

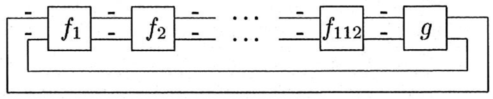

Figure 1: A 1-D CML simulation of a universal Turing machine.

The obvious idea here is to have each CML site k compute the component map σ_k, associated to square $I_k \subseteq [0,1]^2$. The response function f_k transforms nontrivially only the vectors x in I_k, and on those computes the correct affine mapping $x \mapsto A_k x + \lambda_k$; elsewhere the response of site k is the identity.

We make one modification to this straightforward scheme, in order to keep the computations at different sites over one period of the chain from interfering with each other.[1] For this purpose, we have each f_k map the vectors it responds to out of the domains I_j of the other f_j; and then have the response function g at site 113 map the transformed vectors back to the usual range. Precisely, we define the local response functions as follows:

$$f_k(x) = \begin{cases} A_k x + \lambda_k + (\mu,\mu)^T, & \text{if } x \in I_k, \\ x, & \text{otherwise,} \end{cases} \tag{2}$$

and $g(x) = x - (\mu,\mu)^T$, where $\mu > 0$ is some sufficiently large constant. (In the present situation, any $\mu > 1$ suffices, but later we shall need bigger shifts.) We note that the vector-valued response functions f_k are not unimodal; this defect will be corrected in the scalar versions.

We now proceed to show how to simulate the vector-state 1-D CML of Fig. 1 on a scalar-state 2-D CML. Although our eventual goal is to obtain a lattice with symmetric interconnections, it will be convenient to present the construction as if the interconnections were asymmetric. Thus, we first observe that an asymmetric connection from scalar site i to scalar site j, whose states are constrained to some interval $[0,\mu]$, may be simulated by interposing between sites i and j three additional sites a, b, and c, with states in the interval $[0, 3\mu]$, and with the unimodal response functions (cf. Fig. 2):

$$\begin{aligned} f_a(x) &= \begin{cases} x + 2\mu, & \text{if } 0 \le x \le \mu, \\ 0, & \text{otherwise,} \end{cases} \\ f_b(x) &= \begin{cases} x - \mu, & \text{if } 2\mu \le x \le 3\mu, \\ 0, & \text{otherwise,} \end{cases} \\ f_c(x) &= \begin{cases} x - \mu, & \text{if } \mu \le x \le 2\mu, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

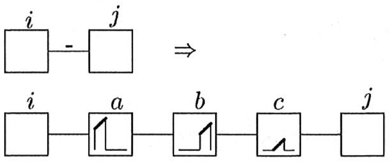

Figure 2: Symmetric simulation of an asymmetric connection.

It can be verified that with this construction, information can only flow from left to right; the influence of site j back on site i is always zero.

Consider then a site with a vector-valued state $(x,y) \in \mathcal{R}^2$ and response function $f(x,y) = (x^+, y^+)$, where

$$x^+ = \begin{cases} \alpha x + \beta y + \lambda + \mu, & \text{if } a \le x \le b, \\ & \quad c \le y \le d, \\ x, & \text{otherwise,} \end{cases}$$

$$y^+ = \begin{cases} \gamma x + \delta y + \nu + \mu, & \text{if } a \le x \le b, \\ & \quad c \le y \le d, \\ y, & \text{otherwise,} \end{cases}$$

with $a,b,c,d \in [0,1]$.

Let us concentrate on the computation of x^+ from x and y. In the construction of equation (2), let us choose the constant μ so large that for all $x, y \in [0,1]$,

$$\begin{aligned} (\alpha-1)x + \lambda + \mu/2 &< 2\mu/3, \\ \beta y + \mu/2 &< 2\mu/3, \\ (\alpha-1)x + \beta y + \lambda + \mu &\ge 2\mu/3. \end{aligned} \tag{3}$$

Since in the case of the Koiran *et al.* simulation, it is always the case that $\alpha x + \beta y + \lambda \in [0,1]$, it in fact suffices to choose here $\mu > 6$.

We shall separate the variables in computing x^+ by first computing independently two unimodal scalar functions $f_A(x)$ and $f_B(y)$, whose results are then combined via a third unimodal function $f_C(z)$ to obtain the value $\Delta x = x^+ - x = f_C(f_A(x) + f_B(y))$. This value is then simply summed with x to obtain x^+. The functions f_A, f_B, and f_C are defined as follows:

$$\begin{aligned} f_A(x) &= \begin{cases} (\alpha-1)x + \lambda + \mu/2, & \text{if } a \le x \le b, \\ 0, & \text{otherwise,} \end{cases} \\ f_B(y) &= \begin{cases} \beta y + \mu/2, & \text{if } c \le y \le d, \\ 0, & \text{otherwise.} \end{cases} \\ f_C(z) &= \begin{cases} 0, & \text{if } z < 2\mu/3, \\ z, & \text{if } z \ge 2\mu/3. \end{cases} \end{aligned}$$

It can be verified that if the value of μ satisfies the conditions (3), one then obtains as $\Delta x = f_C(f_A(x) + f_B(y))$ the value:

$$\Delta x = \begin{cases} (\alpha-1)x + \beta y + \lambda + \mu, & \text{if } a \le x \le b, \\ & \quad c \le y \le d, \\ 0, & \text{otherwise,} \end{cases}$$

i.e. $\Delta x = x^+ - x$, as desired.

The computation of y^+ from x and y may be similarly decomposed into four scalar steps as $y^+ = f_F(f_D(y) + f_E(x)) + y$, with analogously defined functions f_D, f_E, and f_F.

This computation scheme can be implemented as a two-dimensional scalar CML 5 × 9 -site module with asymmetric interconnections, as indicated in Fig. 3. (Symmetrizing the connections with the construction of Fig. 2 then yields a 20 × 33 -site symmetric module.) The upper and lower halves of the module are mirror images of each other, except for the response functions at sites A through E, which are f_A through f_E, respectively. The response functions at sites with no special markings are the identity, except for the dash-marked sites, whose responses are identically zero. The upper half of the module computes the value x^+ and the lower half computes y^+. Let us concentrate on the former computation.

[1]Actually, interference is not a problem, if we don't mind the CML simulating several steps of the Turing machine over one period. However we shall need the construction also for another purpose later, so it is convenient to introduce it here.

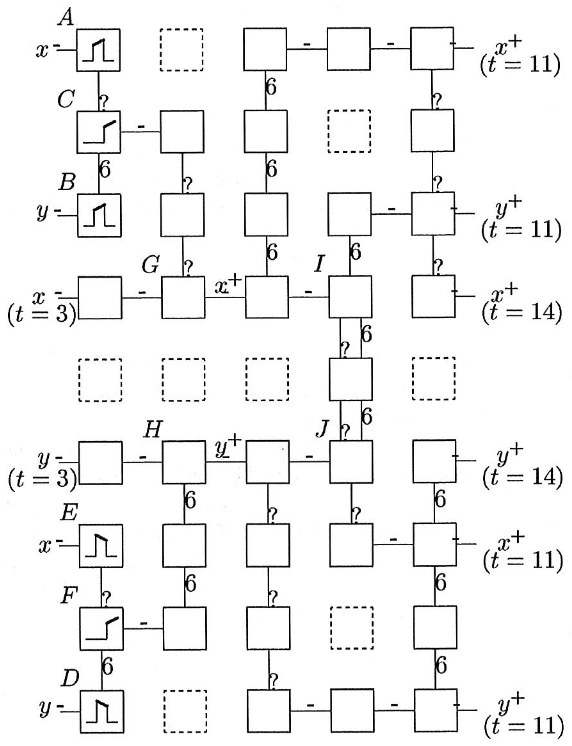

Figure 3: Scalar simulation of a vector-valued site.

Here the sites A and B receive as input at time $t = 0$ the values of x and y from the preceding module. At time[2] $t = 1$ these sites compute the values $f_A(x)$ and $f_B(y)$, which are then combined at time $t = 2$ into $\Delta x = f_C(f_A(x)+f_B(y))$ at site C. This value is then propagated to site G, where it is summed at time $t = 5$ with another copy of value x, delayed for 3 time units in the previous module, to obtain the new value x^+.

A similar computation is performed at sites D–H in the lower half of the module to obtain the new value y^+. The purpose of the remaining sites is then simply to propagate the new values x^+ and y^+ so that they appear as inputs to the following module at the correct locations and with the correct relative delays (specifically, at times $t = 11$ and $t' = 11 + 3 = 14$). Due to the local connectivity restriction, even this simple task requires some care.

A particularly delicate arrangement occurs at sites I and J, where a copy of the value x^+ is transmitted from the upper to the lower half of the module, simultaneously as a copy of the value y^+ is transmitted in the opposite direction. Here we may again apply the basic construction of Fig. 2, but in this case in the extended form of Fig. 4, in order to achieve the simultaneous exchange with the correct time delays.

[2]The delay times are here indicated in terms of the asymmetric lattice. Because of the construction of Fig. 2, the delays in the symmetric lattice are four times longer.

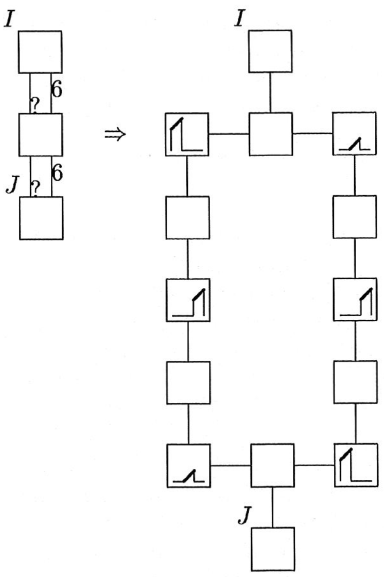

Figure 4: A symmetric state-exchange lattice.

Elsewhere, the correctness of the construction in Fig. 3 can be verified by inspection. One should note, though, that besides the correct values of x^+ and y^+ with the appropriate time delays, also some superfluous intermediate results will emerge from the output sites of the modules. For instance, because of the branching at site I, the value x^+ appears, superfluously, at time $t = 9$ at the same upper-half site that is assigned as the output site for value y^+ at time $t = 11$. However, because of the thresholding at the C and F sites in the following module, these superfluous activations will not propagate further in the lattice.

4 Conclusions

Starting from the iterated-map simulation by Koiran *et al.*[7] of Minsky's[8] small universal Turing machine, we have presented three constructions of CML's capable of universal computation. The simplest construction (Fig. 1) is an asymmetric 1-D CML of 113 periodically connected sites with states from $\mathcal{R}^2$. This lattice simulates one step of the universal Turing machine per 113 time units. The second construction (Fig. 3) implements each vector-valued site of this 1-D CML on a 2-D sublattice of 5×9 scalar-valued sites, resulting in a scalar asymmetric CML of $565 \times 9 = 5085$ sites that simulates one Turing machine step per $11 \times 565 = 6215$ time units. Finally, the asymmetric connections in the 2-D CML may each be simulated by a sequence of four symmetric connections (Fig. 2), yielding a scalar 2-D CML with $2260 \times 33 = 74580$ symmetrically interconnected sites, and with a time dilation factor of 24860 as compared to the original Turing machine.

All our lattices are strongly anisotropic in the sense of having site-dependent response functions—on the other hand, the couplings between sites are purely local, symmetric, and homogeneous.[3]

It remains an open question whether also completely isotropic CML's are capable of universal computation. Also, one should try to improve the efficiency of the current simulation in the direction of making better use of the inherent parallelism of the CML model.

References

[1] ASARIN, Eugene, and Oded MALER, "On some relations between dynamical systems and transition systems", *Proceedings of the 21st International Colloquium on Automata, Languages, and Programming* (Serge ABITEBOUL and Eli SHAMIR eds.), Springer-Verlag (1994), 59–72

[2] DEVANEY, Robert L. *An Introduction to Chaotic Dynamical Systems*, 2nd Edition, Addison-Wesley (1989).

[3] HOLDEN, A. V., J. V. TUCKER, and B. C. THOMPSON, "Can excitable media be considered as computational systems?", *Physica D* **49** (1991), 240–246.

[4] KANEKO, Kunihiko, "Pattern dynamics in spatiotemporal chaos: pattern selection, diffusion of defect and pattern competition intermittency", *Physica D* **34** (1989), 240–246.

[5] KANEKO, Kunihiko, "Clustering, coding, switching, hierarchical ordering, and control in a network of chaotic elements", *Physica D* **41** (1990), 137–172.

[6] KANEKO, Kunihiko, "The coupled map lattice: introduction, phenomenology, Lyapunov analysis, thermodynamics and applications", *Theory and Applications of Coupled Map Lattices* (Kunihiko KANEKO ed.), John Wiley & Sons (1993), 1–49.

[7] KOIRAN, Pascal, Michel COSNARD, and Max GARZON, "Computability with low-dimensional dynamical systems", *Theoretical Computer Science* **132** (1994), 113–128.

[8] MINSKY, Marvin, *Computation: Finite and Infinite Machines*, Prentice-Hall (1967).

[9] MOORE, Cristopher, "Unpredictability and undecidability in dynamical systems", *Physical Review Letters* **64** (1990), 2354–2357.

[10] SIEGELMANN, Hava T., and Eduardo D. SONTAG, "On the computational power of neural nets", *J. Computer and System Sciences* **50** (1995), 132–150.

[3]Curiously, an earlier study of the computational capabilities of CML's and related models[3] came to the following conclusion: "[Our study] emphasizes the computational poverty of excitable media. [...] The only way to extend the computational capabilities of an excitable medium is by radical anisotropy, in which different parts of the medium comprise different components, and also have different connectivities, involving specific non-local connections, as in a neural network." Our results show that while this assessment may be correct as regards the requirement for anisotropy, it is clearly wrong with respect to the connectivities.