# First International Workshop on Theoretical and Experimental Material Computing (TEMC 2019)

Tokyo, Japan
3 June 2019

TEMC 2019 is sponsored by the
EPRSC SpInspired project EP/R032823/1 and the University of York, UK

EPSRC
Engineering and Physical Sciences Research Council

UNIVERSITY of York

The **First International Workshop on Theoretical and Experimental Material Computing** (TEMC 2019) is being held in Tokyo, Japan, as a satellite workshop of the International Conference on Unconventional Computation and Natural Computation (UCNC 2019), 3-7 June 2019.

**Material computing** exploits unconventional physical substrates and/or unconventional computational models to perform physical computation in a non-silicon and/or non-Turing paradigm.

TEMC 2019 encompasses a range of theoretical and experimental approaches to material computing. The aim of the workshop is to bring together researchers from a range of connected fields, to inform of latest findings, to engage across the disciplines, to transfer discoveries and concepts from one field to another, and to inspire new collaborations and new ideas.

After the workshop, there is an open call for a **special issue of the International Journal of Unconventional Computing** on the workshop theme.

**Programme and Organising Committee**

Susan Stepney
Matt Dale
Jack Dewhirst
Simon O'Keefe
Angelika Sebald
Martin Trefzer

**Programme** (3 June 2019)

14:00 Kohei Nakajima, University of Tokyo (keynote speaker)
***Physical reservoir computing for soft robots***

15:00 [coffee break]

15:30 Jonny Edwards, Temporal computing.
***A Measure of Oscillator Purity for Computing with Time***

16:00 Sayed Ahmad Salehi, University of Kentucky.
***DNA-based Machine Learning Using Factorial Coding***

16:30 Odd Rune Lykkebø, Johannes H. Jensen, Arthur Penty, Anders Strømberg, Magnus Själander, Erik Folven, Gunnar Tufte, NTNU.
***Emergent computation on a magnetic lattice***

# Physical reservoir computing for soft robots

**Kohei Nakajima**
University of Tokyo

Soft robots are typical underactuated systems that generate diverse body dynamics when actuated. In this presentation, a technique to exploit these diverse body dynamics directly as an information processing device is introduced, which is called *Physical Reservoir Computing* (PRC). This technique enables us to take physical dynamics from information processing view point. Several physical substrates/platforms for PRC including soft materials (e.g., silicone-based soft robotic arm) are presented, and illustrate the potentials of the framework through a number of experiments. Our focus will particularly be on how dynamical system aspects can provide a novel view to soft robotics through the PRC framework, including the relevance of chaotic dynamics and input-induced phenomena.

# A Measure of Oscillator Purity for Computing with Time

Jonny Edwards

Temporal Computing

## 1 Introduction

The purpose of this short abstract is to describe an initial measure (in a similar vein to [1]) for the suitability of a medium for computing using clocked oscillations. We briefly review the scope for computing in this manner, using a temporally structured unary coding [2]. The key requirement for this approach is pure oscillation, which can be measured using entropy based spectral methods.

## 2 Simple Temporal Wave Operations

Temporal compute methods differ from traditional computation in that the substrate is time. In temporal arithmetic, addition is computed using a concatenation, but differs from the digital implementation described in [2] by using the native oscillation of the signal to clock the output. Multiplication is then the act of scaling the value against a base frequency. See Figure 1 for a visual explanation. Since the primary substrate is time, the major factors in increasing the speed and accuracy of computation are the oscillator frequency and purity.

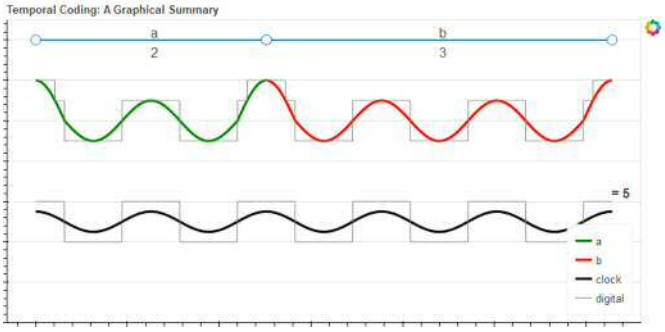



**Fig. 1.** The addition of 2+3 in a temporal wave context, we use concatenation and re-parametrisation to perform the arithmetic.

To measure clock correctness (clockness) we perform a simple frequency analysis using the Fourier transform $Z \int_{-\infty}^{\infty} x(t)e^{-jwt}dt \in C$, examining the magnitude(a) ($a = \sqrt{(}Re(x)^2 + Im(x)^2)$). For a single oscillation, this process gives

the relative presence of that oscillation, which can be easily normalised by the total power of the spectrum ($N(a_n) = a_n / \sum_i^{maxfreq} a_i$). This transforms the spectrum into a probability mass function, which gives the probability of oscillation at a specific frequency. To turn this into a measure is then straightforward, we examine the entropy of the spectrum and **a perfect clock will have zero entropy**. The entropy is given by:

$$H(X) = -\sum_{x \in X} p(x) \lg p(x) \tag{1}$$

This is often referred to as *spectral entropy*, but here we are using it specifically to characterise clocks. Figure 2 provides some simple examples.

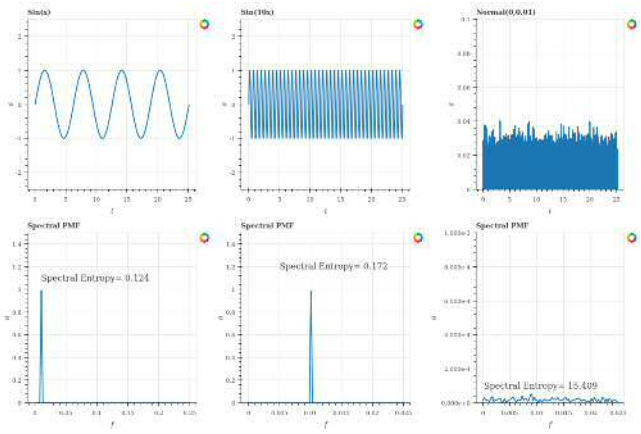



**Fig. 2.** Some example functions assessed for their clockness, sample rate was 10000 points, with some approximation due to resolution

The above scheme is used to evaluate the specific clock property of an oscillating medium, to assess suitability of the medium to conduct temporally organised compute.

## References

1. Dale, M., Miller, J.F., Stepney, S., Trefzer, M.A.: A substrate-independent framework to characterise reservoir computers. CoRR **abs/1810.07135** (2018), http://arxiv.org/abs/1810.07135
2. Edwards, J., O'Keefe, S., Henderson, W.D.: Unconventional arithmetic: A system for computation using action potentials. In: Proc. of Unconventional Computation and Natural Computation. pp. 155–163 (2014)

# DNA-based Machine Learning Using Factorial Coding

*Sayed Ahmad Salehi*

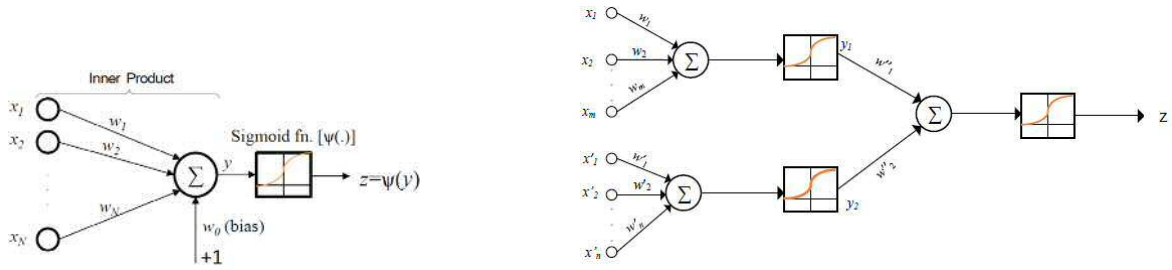
*Electrical and Computer Engineering Department,*
*University of Kentucky, Lexington, KY, USA*

The field of synthetic biology has advanced remarkably in the last 20-25 years. The progress in the field continues to accelerate at a rate even faster than Moore's law that refers to doubling in the number of transistors on an integrated circuit (IC) chip every 18 months. Due to these advances, biomolecular computing, particularly DNA computing, has emerged as a non-conventional computing technology. Just as electronic systems implement computation in terms of voltage, biomolecular systems compute in terms of chemical concentrations (*amount of molecules per unit volume*). A broad range of computational processes has been considered for molecular implementation. In this presentation, we briefly demonstrate the development of molecular systems for performing machine learning algorithms particularly multi-level perceptron networks.

Here, we present molecular machine learning and DNA-based artificial neural networks. Since machine learning classifiers have become increasingly ubiquitous in different fields of computational science, their physical realization using different technologies has been considered. Accordingly, molecular implementations of machine learning classifiers and artificial neural networks have been considered by researchers. The goal is to design biomolecular systems that are programmed to learn from their interaction with the living cells and decide when and how to act appropriately. For example, genetic logic circuits for cell classification can sense features of mRNAs; they can detect their expression patterns and selectively respond to specific cell types. Such circuits could enable the production of personalized smart drugs that target specific diseases for specific patients. In general, an artificial neural networks consists of one or more layers where, in each layer, a neuron computes a weighted sum followed by a nonlinear activation (transfer) function. Typically, the activation function corresponds to a sigmoid function.

We illustrate the implementation of a simple neural network, i.e., single-layer perceptron, presented in prior work [1]. The implementation is based on fractional coding proposed by Salehi et al. [2]. Conventional coding in molecular computing represents each variable directly by the concentration of a molecular type, e.g., for variable $x$ we have $x = [X]$ , where $[.]$ denotes concentration. Fractional coding, however, uses the ratio of two molecular concentrations to represent each variable, e.g., $x = \frac{[X_1]}{[X_0]+[X_1]}$. Interestingly, fractional coding relates molecular computing to stochastic computing in electronics, where variables are represented by random bit streams. Accordingly, computations performed by stochastic computing circuits can be mapped to molecular computing structures based on fractional coding. As it is shown in Figure 1 (a), the first step to implement a perceptron with $N$ inputs is to calculate $x_i \times w_i$ for $i = 1,2, \dots, N$. Using fractional coding, each multiplication can be implemented by four simple molecular reactions. The second step is to calculate the sum of the multiplication results. In molecular implementation, the sum can be calculated with no extra cost; if all the reactions calculating the multiplications in the first step, produce the same product, the product would represent the desired sum. The last step is the activation function. Although prior work on molecular implementation of neural networks has considered either a hard-threshold or linear transfer function as the activation function, in real-word applications sigmoid function is a common activation function. We explain how this function could be approximately computed by molecular reactions based on fractional coding.

Rather than the sum, molecular reactions calculate the scaled sum of the inputs, i.e., $\frac{1}{N}\sum_{i=1}^{N}(x_i \times w_i)$. Where $N$ is a large number, the output is scaled down too much and could lead to a wrong classification decision. To address this problem we propose a new multi-layer perceptron neural network. Figure 1(b) shows a two-layer version of the proposed network. The network consists of three single-layer perceptron units. Obviously, the number of inputs for each unit is less compared to when using only one single-layer perceptron unit. The network provides flexibility to adjust the number of inputs for each unit and thus its scaling factor. By cascading more perceptron units, the structure can be generalized for multi-layer networks with different hidden layers.




(a) (b)

Figure 1. (a): A single-layer perceptron, (b) Proposed multi-layer perceptron.

We use chemical reaction networks (CRNs) as the programming language to design our computing molecular systems and then map them to DNA reactions based on the DNA strand-displacement mechanism. We then validate the DNA implementation of these designs by showing their simulation results.

[1] Salehi, S.A., Liu, X., Riedel, M.D., Parhi, K.K.: Computing mathematical functions using DNA via fractional coding. Sci. Rep. 8(8312), Nature.com, (2018)

[2] Salehi, S. A., Parhi, K. K. & Riedel, M. D. Chemical Reaction Networks for Computing Polynomials. ACS Synthetic Biology Journal, 6(1), 76-83, Jan. (2017).

# Emergent computation on a magnetic lattice

Odd Rune Lykkebø, Johannes H. Jensen, Arthur Penty, Anders Strømberg,
Magnus Själander, Erik Folven and Gunnar Tufte
firstname.lastname@ntnu.no

Norwegian University of Science and Technology (NTNU)

Artificial spin ice (ASI) are systems of coupled nanomagnets with interesting computational properties. These magnetic systems consist of large numbers of nonlinear elements arranged on a 2D lattice whose local interactions result in complex global behaviors. We have demonstrated interesting dynamics in small arrays of 40 nanomagnets through micromagnetic simulations [2]. Here, we present results for much larger arrays with thousands of interacting magnets. Large scale simulations based on dipolar interactions [1] enable us to efficiently study higher order properties, such as robustness, memory and collective dynamics.

## 1 Artificial spin ice

In ASI, the shape of the nanomagnets is chosen such that they all have well-defined magnetization direction, i.e., each magnet has a binary state. A magnet can switch state if subjected to a sufficiently strong magnetic field, either coming from neighboring magnets or from an applied external field. The interactions depend crucially on the geometric arrangement of the lattice, e.g., the spacing and angles between magnets determine the nature of their interactions.

Figure 1 shows two ASI systems, schematically in black, and their large scale simulated behaviors in color. The system on the left is a 'square spin ice' [6]. The square geometry results in a system with a high degree of frustration, i.e., where all competing interactions cannot be satisfied at the same time. Notice the large demagnetized regions (white) separated by magnetized borders (colored).

By rotating each element by 45° around its center, as shown in the right figure, we get the so-called 'pinwheel spin ice' [4]. This small change leads to qualitatively different behavior. Notice the large magnetically ordered domains (filled colored regions).

## 2 Computing in artificial spin ice

Computation and memory is closely coupled in these nanomagnetic systems. The magnetization of each nanomagnet is essentially a non-volatile memory. Magnetic dipolar coupling between the magnets results in a network of large numbers of interacting nonlinear elements.

The system dynamics can be readily tuned by changing parameters such as spacing between the magnets, the size of magnets, external fields and different lattice geometries. We find interesting dynamic regimes ranging from fixed points,

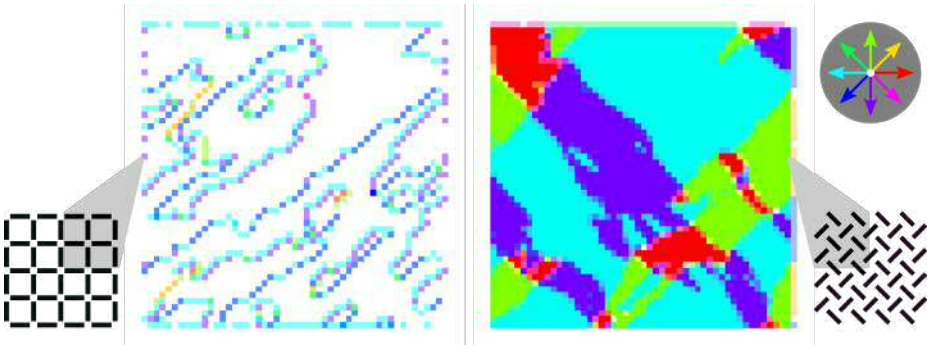


Fig. 1: Simulations of 100x100 artificial spin ice. *Left*: Square spin ice favors demagnetized regions separated by bands of magnetized borders. *Right*: Pinwheel spin ice has long-range magnetic ordering, resulting in large magnetized regions. Both systems have a total of ∼20k magnets. Each pixel represents the net magnetization of eight magnets. The color indicates magnetization direction as shown in the color map, where white represents zero net magnetization.

limit cycles and (very) long trajectories. In the fixed point regime, the system could be a natural fit for ballistic forms of computation [5]. Long trajectories, on the other hand, are more suitable for trajectory-based computing paradigms such as reservoir computing [3].

Because of the bistable nature of each magnet, the system is naturally resilient to noise [6]. Beyond simulations we have started to manufacture large magnetic arrays in the lab. Readout of individual magnets is still confined to the lab, but observing larger regions of magnetization has the potential for a desktop device. Our simulations show that manufacturing imperfections are useful and even necessary to obtain interesting dynamics.

## References


1. Budrikis, Z.: Athermal dynamics of artificial spin ice: disorder, edge and field protocol effects. Ph.D. thesis, The University of Western Australia (2012)
2. Jensen, J.H., Folven, E., Tufte, G.: Computation in artificial spin ice. In: The 2018 Conference on Artificial Life. pp. 15–22. MIT Press, Cambridge, MA (2018)
3. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural computation **14**(11), 2531–2560 (2002)
4. Macêdo, R., Macauley, G.M., Nascimento, F.S., Stamps, R.L.: Apparent ferromagnetism in the pinwheel artificial spin ice. Physical Review B **98**, 014437 (2018)
5. Stepney, S.: The neglected pillar of material computation. Physica D: Nonlinear Phenomena **237**(9), 1157–1164 (2008)
6. Wang, R.F., Nisoli, C., Freitas, R.S., Li, J., McConville, W., Cooley, B.J., Lund, M.S., Samarth, N., Leighton, C., Crespi, V.H., Schiffer, P.: Artificial "spin ice" in a geometrically frustrated lattice of nanoscale ferromagnetic islands. Nature **439**, 303–306 (2006)