

# Fractalization

Tanaka Shuma

(Dated: December 3, 2025)

## Abstract

There are two types of fracton phases. Type-I fracton models may host excitations with restricted mobility, such as fractons, but they also allow for the existence of partially mobile excitations, such as lineons. An example is the X-cube model. In contrast, the Type-II fracton model is a phase where all excitations are fractons, and Haah's code is an example."Fractalization" is a procedure that maps an operator defined on a  $D$ -dimensional lattice to an operator on a  $D + m$ -dimensional lattice, taking as input a set of  $m$ -dimensional linear cellular automaton (LCA) rules. This operation allows for the interpretation of Type-II fracton phases, fractal symmetry-protected topological (SPT) phases, and more, in terms of well-understood lower-dimensional spin models.

## CONTENTS

I. Introduction:Linear cellular automata	2
II. Fractalization	6
A. Definition	6
B. Properties	7
1. Locality	7
2. Commutativity	8
III. Examples	8
A. 1D Ising model	8
B. 1D Cluster model	8
C. 2D Toric code model	8
References	8

## I. INTRODUCTION: LINEAR CELLULAR AUTOMATA

By using linear cellular automata(LCA), we can fractalize the system written in stabilizer code. Let  $c_{r,t} \in \{0, 1\}$  represent the state of cell  $\mathbf{r} = (r_1, \dots, r_D) \in Z^D$  of a cellular automaton at time  $t$ . The time evolution of LCA is defined as linear function (mod 2) of the previous state. In this note, we focus on the LCA which satisfy the following property.

- (First order)  $c_t$  is determined only by  $c_{t-1}$
- (Locality)  $F_{\mathbf{r},\mathbf{r}'}$  becomes nonzero only when  $|\mathbf{r} - \mathbf{r}'|$  is small.
- (Transformation invariance)  $F_{\mathbf{r},\mathbf{r}'} \equiv f_{\mathbf{r}-\mathbf{r}'}$  only depends on the difference  $\mathbf{r} - \mathbf{r}'$ .

where  $F$  matrix is defined as follows;

$$c_{\mathbf{r},t} = \sum_{\mathbf{r}'} F_{\mathbf{r},\mathbf{r}'} c_{\mathbf{r}',t-1} \equiv \mathbf{F} c_{t-1}. \quad (1)$$

We give the starting state  $c_0$  at  $t = 0$ , and the spacetime trajectory at all future is completely given as

$$c_t = \mathbf{F}^t c_0. \quad (2)$$

We adopt polynomial representation of the state  $c_t$ . If the space dimension is  $D$ ,  $c_t$  is represented as  $D$ -variate polynomial with  $\mathbb{F}_2$  coefficients.

$$c_t(x) = \sum_{\mathbf{r}} c_{\mathbf{r},t} \mathbf{x}^{\mathbf{r}} \in \mathbb{F}_2[\mathbf{x}]. \quad (3)$$

where  $\mathbf{x} = \{x_i\}_{i \in [1, \dots, D]}$ ,  $\mathbf{x}^{\mathbf{r}} := \prod_i x_i^{r_i}$  are monomials. Updating rule by  $\mathbf{F}$  is also represented as

$$f(\mathbf{x}) = \sum_{\mathbf{r}} f_{\mathbf{r}} \mathbf{x}^{\mathbf{r}}. \quad (4)$$

By using this representation, (1) and (2) are easily written as

$$c_t(\mathbf{x}) = f(\mathbf{x}) c_{t-1}(\mathbf{x}) = f(\mathbf{x})^t c_0(\mathbf{x}). \quad (5)$$

From now on, we will switch between the vector representation and the polynomial representation depending in the context.

We always assume that  $f(\mathbf{x})$  has non-zero constant term, and only positive powers of  $x_i$ , and is non-trivial ( $f(\mathbf{x}) \neq 1$ ).

Why LCAs generate the fractal structure? To see this, consider the time  $t = 2^n$ . In the  $\mathbb{F}_2$  polynomials, the coefficients of

$$(a + b)^{2^n} = \sum_k \binom{2^n}{k} a^{2^n-k} b^k \quad (6)$$

become zero when  $k = 1, \dots, 2^n - 1$ , so it is written as

$$(a + b)^{2^n} = a^{2^n} + b^{2^n}. \quad (7)$$

$f(\mathbf{x})^{2^n}$  will be

$$f(\mathbf{x})^{2^n} = f(\{x_i\})^{2^n} = \left( \sum_{\mathbf{r}} f_{\mathbf{r}} \mathbf{x}^{\mathbf{r}} \right)^{2^n} = \sum_{\mathbf{r}} f_{\mathbf{r}} (\mathbf{x}^{\mathbf{r}})^{2^n} = \sum_{\mathbf{r}} f_{\mathbf{r}} (\mathbf{x}^{2^n})^{\mathbf{r}} = f(\mathbf{x}^{2^n}) = f(\{x_i^{2^n}\}). \quad (8)$$

This suggests

$$c_t(\mathbf{x}) = f(\mathbf{x})^{2^n} c_0(\mathbf{x}) = f(\mathbf{x}^{2^n}) c_0(\mathbf{x}). \quad (9)$$

In other words, the time-evolved state consists of multiple replicas of the initial configuration placed at positions  $2^n a_j$ , where  $a_j$  runs over the exponents of  $f(\mathbf{x})$ . Because the relative positions scale as power of two, the pattern generated by the LCA becomes self-similar at exponentially separated length scales, producing a fractal support. In fractalization, that time evolution is reinterpreted as a spatial direction (a new dimension), so instead of flickering in out, the fractal pattern shows up uniformly and simultaneously as a static spatial structure.

As usual, we consider LCAs on finite  $(L_1, \dots, L_d)$  systems with periodic boundary condition. This is implemented by the condition  $x_i^{L_i} = 1$ .

One of the important property of LCA is the reversibility for a given system size. The condition is the exsistance of  $f(\mathbf{x})^{-1} \in \mathbb{F}_2[\mathbf{x}]$  that satisfies  $f(\mathbf{x})^{-1} f(\mathbf{x}) = 1$ . A simple example of reversible LCA is given by any polynomial satisfying  $f(1) = 1$  with system size  $L_i = 2^n$ . This follows the fact that  $f(\mathbf{x})^{2^n} = f(\mathbf{x}^{2^n}) = f(\mathbf{x}^{L_i}) = f(1) = 1$ , so the inverse is given by  $f(\mathbf{x})^{-1} = f(\mathbf{x})^{2^n-1}$ .

Let us consider one example with  $d = 1, f(x) = 1 + x$ . Updating rule will be

$$c_t(x) = (1 + x)c_{t-1}(x) = \sum_r c_{r,t-1} x^r + \sum_r c_{r,t-1} x^{r+1} = \sum_r c_{r,t} x^r \quad (10)$$

This is equal to

$$c_{r,t} = c_{r,t-1} + c_{r-1,t-1} = (\delta_{r,r'} + \delta_{r,r'+1}) c_{r,t-1}. \quad (11)$$

The initial state  $c_0(x) = 1$  (corresponding to  $c_{\mathbf{r},0} = \delta_{\mathbf{r},0}$ ). Updating rule for several terms are

$$\begin{aligned}
 f(x)^0 &= 0 \\
 f(x)^1 &= 1 + x \\
 f(x)^2 &= 1 + 2x + x^2 = 1 + x^2 \\
 f(x)^3 &= (1 + x)(1 + x^2) = 1 + x^2 + x + x^3 = 1 + x + x^2 + x^3 \\
 f(x)^4 &= (1 + x)(1 + x + x^2 + x^3) = 1 + x^4 \\
 &\dots
 \end{aligned} \tag{12}$$

The fractal generated is the Sierpinski triangle (Pascal's triangle mod 2).  $f(x) = 1 + x$  cannot be reversive on any system size. A simple example of a reversible LCA is  $f(x) = 1 + x + x^2$  on sizes  $L = 2^n$ . (This satisfies  $f(1) = 1$ .)

We can use this LCAs procedure to extend CSS quantum code, Let us consider qubits on the site  $\mathbf{r}$  of a hypercube lattice.  $X_{\mathbf{r}}$  and  $Z_{\mathbf{r}}$  means Pauli operators located on  $\mathbf{r}$ . Define a map  $\sigma_X$  which maps a binary vector  $\mathbf{a} = (a_{\mathbf{r}})$  to a tensor product of  $X$  Pauli operator as

$$\sigma_X[\mathbf{a}] = \prod_{\mathbf{r}} X_{\mathbf{r}}^{a_{\mathbf{r}}}. \tag{13}$$

Similarly,  $\sigma_Z$  is defined as

$$\sigma_Z[\mathbf{b}] = \prod_{\mathbf{r}} Z_{\mathbf{r}}^{b_{\mathbf{r}}}. \tag{14}$$

Commutation relation between  $\sigma_X$  and  $\sigma_Z$  are calculated as

$$[[\sigma_X[\mathbf{a}], \sigma_Z[\mathbf{b}]]] = (-1)^{\mathbf{b}^T \mathbf{a}} \tag{15}$$

where  $[[\cdot, \cdot]]$  is group commutator.

$$[[A, B]] = A^{-1} B^{-1} A B. \tag{16}$$

Let us show Eq.(15). Group commutator of Pauli operators are

$$[[\sigma_X[\mathbf{a}], \sigma_Z[\mathbf{b}]]] = \sigma_X[\mathbf{a}] \sigma_Z[\mathbf{b}] \sigma_X[\mathbf{a}] \sigma_Z[\mathbf{b}] \tag{17}$$

because Pauli operators are involution  $X^{-1} = X, Z^{-1} = Z$ .  $\sigma_X[\mathbf{a}] \sigma_Z[\mathbf{b}]$  are calculated as

$$\sigma_X[\mathbf{a}] \sigma_Z[\mathbf{b}] = \prod_{\mathbf{r}} X_{\mathbf{r}}^{a_{\mathbf{r}}} \prod_{\mathbf{r}'} Z_{\mathbf{r}'}^{b_{\mathbf{r}'}} = \prod_{\mathbf{r}} X_{\mathbf{r}}^{a_{\mathbf{r}}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} = (-1)^{\sum_{\mathbf{r}} a_{\mathbf{r}} b_{\mathbf{r}}} \prod_{\mathbf{r}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} X_{\mathbf{r}}^{a_{\mathbf{r}}} = (-1)^{\mathbf{b}^T \mathbf{a}} \prod_{\mathbf{r}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} X_{\mathbf{r}}^{a_{\mathbf{r}}} \tag{18}$$

Then (17) becomes

$$\sigma_X[\mathbf{a}]\sigma_Z[\mathbf{b}]\sigma_X[\mathbf{a}]\sigma_Z[\mathbf{b}] = \prod_{\mathbf{r}} X_{\mathbf{r}}^{a_{\mathbf{r}}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} X_{\mathbf{r}}^{a_{\mathbf{r}}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} = (-1)^{\mathbf{b}^T \mathbf{a}} \prod_{\mathbf{r}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} X_{\mathbf{r}}^{a_{\mathbf{r}}} X_{\mathbf{r}}^{a_{\mathbf{r}}} Z_{\mathbf{r}}^{b_{\mathbf{r}}} = (-1)^{\mathbf{b}^T \mathbf{a}}. \quad (19)$$

Therefore Eq.(15) is obtained.

We can consider the polynomial representation.  $\mathbf{a} = (a_{\mathbf{r}})$  is correspond to  $a(\mathbf{x}) = \sum_{\mathbf{r}} a_{\mathbf{r}} \mathbf{x}^{\mathbf{r}} \in \mathbb{F}_2[\mathbf{x}]$ . We can think  $\sigma_{X/Z}$  as the map from  $\mathbb{F}_2[\mathbf{x}]$  to Pauli operators. Translation are expressed as multiplying a monomial. By shifting  $\mathbf{r}$ ,  $\sigma_X$  will become

$$\sigma_X[a(\mathbf{x})] \rightarrow \sigma_X[\mathbf{x}^{\mathbf{r}} a(\mathbf{x})]. \quad (20)$$

Including translation, commutation relation are given as follows;

$$[[\sigma_X[\mathbf{x}^{\mathbf{r}} a(\mathbf{x})], \sigma_Z[\mathbf{x}^{\mathbf{r}'} b(\mathbf{x})]] = (-1)^{c_{\mathbf{r}' - \mathbf{r}}}, \quad (21)$$

where

$$c(\mathbf{x}) = \sum_{\mathbf{r}} c_{\mathbf{r}} x^{\mathbf{r}} = a(\mathbf{x}) b(\bar{\mathbf{x}}) \quad (22)$$

and  $\bar{\mathbf{x}} = \mathbf{x}^{-1}$ .

$$\begin{aligned} [[\sigma_X[\mathbf{x}^{\mathbf{r}} a(\mathbf{x})], \sigma_Z[\mathbf{x}^{\mathbf{r}'} b(\mathbf{x})]] &= \sigma_X[\mathbf{x}^{\mathbf{r}} a(\mathbf{x})] \sigma_Z[\mathbf{x}^{\mathbf{r}'} b(\mathbf{x})] \sigma_X[\mathbf{x}^{\mathbf{r}} a(\mathbf{x})] \sigma_Z[\mathbf{x}^{\mathbf{r}'} b(\mathbf{x})] \\ &= \prod_{\mathbf{u}} X_{\mathbf{u}+\mathbf{r}}^{a_{\mathbf{u}}} Z_{\mathbf{u}+\mathbf{r}'}^{b_{\mathbf{u}}} X_{\mathbf{u}+\mathbf{r}}^{a_{\mathbf{u}}} Z_{\mathbf{u}+\mathbf{r}'}^{b_{\mathbf{u}}} \\ &= (-1)^{\sum_{\mathbf{u}} a_{\mathbf{u}} b_{\mathbf{u}+\mathbf{r}-\mathbf{r}'}} \prod_{\mathbf{u}} Z_{\mathbf{u}+\mathbf{r}'}^{b_{\mathbf{u}}} X_{\mathbf{u}+\mathbf{r}}^{a_{\mathbf{u}}} X_{\mathbf{u}+\mathbf{r}}^{a_{\mathbf{u}}} Z_{\mathbf{u}+\mathbf{r}'}^{b_{\mathbf{u}}} \\ &= (-1)^{\sum_{\mathbf{u}} a_{\mathbf{u}} b_{\mathbf{u}+\mathbf{r}-\mathbf{r}'}} \end{aligned} \quad (23)$$

$$\begin{aligned} c(\mathbf{x}) &= \sum_{\mathbf{r}} c_{\mathbf{r}} x^{\mathbf{r}} = a(\mathbf{x}) b(\bar{\mathbf{x}}) = \sum_{\mathbf{u}} a_{\mathbf{u}} \mathbf{x}^{\mathbf{u}} \sum_{\mathbf{u}'} b_{\mathbf{u}'} \mathbf{x}^{-\mathbf{u}'} = \sum_{\mathbf{u}, \mathbf{u}'} a_{\mathbf{u}} b_{\mathbf{u}'} \mathbf{x}^{\mathbf{u}-\mathbf{u}'} = \sum_{\mathbf{u}, \mathbf{k}} a_{\mathbf{u}} b_{\mathbf{u}-\mathbf{k}} \mathbf{x}^{\mathbf{k}} \\ &= \sum_{\mathbf{k}} \left( \sum_{\mathbf{u}} a_{\mathbf{u}} b_{\mathbf{u}-\mathbf{k}} \right) \mathbf{x}^{\mathbf{k}} \end{aligned} \quad (24)$$

For the  $\mathbf{k} = \mathbf{r}' - \mathbf{r}$ ,

$$\sum_{\mathbf{u}} a_{\mathbf{u}} b_{\mathbf{u}-\mathbf{k}} = \sum_{\mathbf{u}} a_{\mathbf{u}} b_{\mathbf{u}-\mathbf{r}'+\mathbf{r}} = \sum_{\mathbf{u}} a_{\mathbf{u}} b_{\mathbf{u}+\mathbf{r}-\mathbf{r}'} = c_{\mathbf{r}'-\mathbf{r}} \quad (25)$$

If we consider more than one qubits in one site of lattice, the polynomial  $a(\mathbf{x})$  will be higher-dimensional. We write  $\mathbf{a}(\mathbf{x}) := \{a_n(\mathbf{x})\}_{n=1, \dots, N}$  where there is  $N$  qubits per site. If we use vectorical representation,  $\sigma$  takes an input  $N$  vectors. (or  $N$  polynomials in polynomial representation.) By calculation, we can obtain commutation polynomial in  $N$  qubits:  $\mathbf{c}(\mathbf{x}) = \mathbf{a}(\mathbf{x})^T \mathbf{b}(\bar{\mathbf{x}})$ .

## II. FRACTALIZATION

### A. Definition

Let us define fractalization of the system. fractalization is the map that maps  $D$ -dimensional  $X$  or  $Z$  Pauli operator to one in  $D + m$  dimension where  $m$  is the dimensional of LCAs. We define  $\mathbf{f}(\mathbf{y}) := \{f_i(\mathbf{y})\}_{i \in [1, \dots, D]}$ ,  $\mathbf{y} = \{y_j\}_{j \in [1, \dots, m]}$ .

Let  $A$  be a local  $X$  Pauli operator in  $D$  dimensions.  $A$  is represented by polynomial as

$$A = \sigma_X[\mathbf{x}^{\mathbf{r}_0} \mathbf{a}(\mathbf{x})]. \quad (26)$$

Here, we chose anchor point  $\mathbf{r}_0$  to maximize each  $(\mathbf{r}_0)_i$  subject to the requirement that  $\mathbf{a}(\mathbf{x})$  has only positive powers of  $x_i$ . Physically,  $\mathbf{r}_0$  means the position of  $A$  operator.

We map this to  $D + m$  dimension. Let  $\mathbf{s}$  denote the position vector along the last  $m$  dimensions. We can specify the position of the site by  $(\mathbf{r}, \mathbf{s})$ . For given  $\mathbf{s}_0$ , the fractalized operator  $A_{\mathbf{s}_0}^{\text{frac}}$  is given by

$$A_{\mathbf{s}_0}^{\text{frac}} = \sigma_X[\mathbf{x}^{\mathbf{r}_0} \mathbf{y}^{\mathbf{s}_0} \mathbf{a}(\mathbf{f}(\mathbf{y}) \circ \mathbf{x})] \quad (27)$$

where

$$\mathbf{f}(\mathbf{y}) \circ \mathbf{x} := \{f_i(\mathbf{y})x_i\}_{i \in [1, \dots, D]} \quad (28)$$

Thus, fractalization maps  $D$ -dimensional operator  $A$  to  $D+m$ -dimensional operators  $\{A_{\mathbf{s}_0}^{\text{frac}}\} : A \mapsto \{A_{\mathbf{s}_0}^{\text{frac}}\}$ . There will be  $A_{\mathbf{s}}^{\text{frac}}$  in each shift  $\mathbf{s}$ . Most of examples will be explained by  $m = 1$  new dimension. Similarly, we can define  $Z$  operator code

$$B = \sigma_Z[\mathbf{x}^{\mathbf{r}_0} \mathbf{b}(\bar{\mathbf{x}})] \quad (29)$$

$\mathbf{r}_0$  is similarly defined, but spatially inversed due to  $\bar{\mathbf{x}}$ .  $(\mathbf{r}_0)_i$  is defined to be minimized in condition that  $\mathbf{b}(\bar{\mathbf{x}})$  contains only positive powers of  $\bar{\mathbf{x}}$ . Then fractalization of  $B$  is written as

$$B_{\mathbf{s}_0}^{\text{frac}} = \sigma_Z[\mathbf{x}^{\mathbf{r}_0} \mathbf{y}^{\mathbf{s}_0} \mathbf{b}(\mathbf{f}(\bar{\mathbf{y}}) \circ \bar{\mathbf{x}})]. \quad (30)$$

We used polynomial representation, but we can also use vectorical representation. LCA rule is given by  $\{\mathbf{F}_i = F_{\mathbf{s}, \mathbf{s}'}^{(i)}\}_{i \in [1, \dots, D]}$ .  $X$ -code is given by

$$A = \sigma_X[a_{\mathbf{r}, n}] \quad (31)$$

and anchor point  $(\mathbf{r}_0)_i$  is defined to be  $a_{\mathbf{r},n} = 0$  for any  $(\mathbf{r} - \mathbf{r}_0)_i < 0$ .  $A_{\mathbf{s}_0}^{\text{frac}}$  is written as

$$\begin{aligned} A_{\mathbf{s}_0}^{\text{frac}} &= \sigma_X[a_{(\mathbf{r},\mathbf{s}),n}^{\text{frac}}] \\ &= \sigma_X\left[\left(\prod_i \mathbf{F}_i^{(\mathbf{r}-\mathbf{r}_0)_i}\right)_{\mathbf{s},\mathbf{s}_0} a_{\mathbf{r},n}\right]. \end{aligned} \quad (32)$$

Due to the translation symmetry all  $\mathbf{F}_i$  commute, so the order of the product  $\mathbf{F}_i$  is arbitrary. (This is also represented as  $f_i(x)f_j(x) = f_j(x)f_i(x)$ .) Similarly,  $Z$ -code is defined as

$$B = \sigma_Z[b_{\mathbf{r},n}] \quad (33)$$

and

$$B_{\mathbf{s}_0}^{\text{frac}} = \sigma_Z\left[\left(\prod_i \mathbf{F}_i^{(\mathbf{r}_0-\mathbf{r})_i}\right)_{\mathbf{s}_0,\mathbf{s}} b_{\mathbf{r},n}\right], \quad (34)$$

where  $(\mathbf{r}_0)_i$  is defined to be  $b_{\mathbf{r},n} = 0$  for  $(\mathbf{r}_0 - \mathbf{r})_i < 0$ .

Given a CSS stabilizer group  $\mathcal{S} = \langle \{\mathcal{O}_l\} \rangle$  and Hamiltonian

$$H = - \sum_l \mathcal{O}_l \quad (35)$$

where  $\mathcal{O}_l$  are local  $X$  or  $Z$  generators of the stabilizer group. By applying fractalization to the Hamiltonian, we obtain

$$H^{\text{frac}} = - \sum_{\mathbf{s},l} \mathcal{O}_{\mathbf{s},l}^{\text{frac}}. \quad (36)$$

This Hamiltonian inherits many of properties of  $H$ . This Hamiltonian is translation invariant along the final  $m$  dimensions.

## B. Properties

### 1. Locality

Fractalization is locality preserving. As long as  $A$  has support only a local patch in  $D$ -dimensional lattice,  $A_{\mathbf{s}_0}^{\text{frac}}$  will also have support on a local patch of the  $D+m$ -dimensional lattice near  $\mathbf{s}_0$ . This is derived from the locality of LCA rule. ( $F_{\mathbf{s},\mathbf{s}_0}^{(i)}$  becomes nonzero only when  $|\mathbf{s} - \mathbf{s}_0|$ .) Non-local operators map on to non-local. let us consider  $D = 1$  chain with length  $L$  and open boundary condition, and utilize 1-dimensional LCA rule  $f(y) = 1 + y$ . The non-local operator

$$S = \prod_r X_r = \sigma_X\left[\sum_r x^r\right] \quad (37)$$

maps on to the fractal

$$S_s^{\text{frac}} = \sigma_X \left[ y^s \sum_r f(y)^r x^r \right] = \sigma_X \left[ y^s \sum_r (1+y)^r x^r \right] \quad (38)$$

in two dimensions.  $f(y)^n$  generates the  $n$ -th row of the Sierpinski triangle fractal. Thus,  $S_s^{\text{frac}}$  is operator that acts as  $X$  on sites along a Sierpinski triangular fractal subsystem. Here  $\mathbf{s}$  denotes an overall shift in the  $y$  direction.

## 2. Commutativity

Fractalization preserves commutativity. Suppose we have two operators  $A = \sigma_X[\mathbf{x}^{\mathbf{r}_0} \mathbf{a}(\mathbf{x})]$  and  $B = \sigma_Z[\mathbf{x}^{\mathbf{r}_1} \mathbf{b}(\bar{\mathbf{x}})]$  satisfying  $[A, B] = 0$ . Then,  $[A_{s_0}^{\text{frac}}, B_{s_1}^{\text{frac}}] = 0$  for all  $\mathbf{s}_0, \mathbf{s}_1$ . This can be seen by calculating commutation polynomial.  $c(\mathbf{x})$  is calculated by

$$c(\mathbf{x}) = \mathbf{x}^{\mathbf{r}_0 - \mathbf{r}_1} \mathbf{a}(\mathbf{x})^T \mathbf{b}(\mathbf{x}). \quad (39)$$

$[A, B] = 0$  holds if and only if  $c(\mathbf{x})$  has zero constant term.

## III. EXAMPLES

### A. 1D Ising model

### B. 1D Cluster model

### C. 2D Toric code model

- 
- [1] T. Devakul and D. J. Williamson, Fractalizing quantum codes, Quantum 5, 10.22331/q-2021-04-22-438, arXiv:2009.01252 (2021).
- [2] M. E. J. Newman and Cristopher Moore, Glassy dynamics and aging in an exactly solvable spin model, Phys. Rev. E 60, 5068(1999).