

It from bit — a concrete attempt

Alexandre Furtado Neto*

January 29, 2023

ORCID 0000-0001-9435-6566

UNESP Alumnus

Abstract

All known lost or incomplete universal symmetries, i.e., left/right parity, matter/antimatter, and white/dark matter are restored or completed on a global level in a finite, closed, discrete Euclidean space. The known fundamental particles emerge from a myriad of even more fundamental entities called bubbles. The electric charge uses one bit, while weak uses two bits and the color force uses three bits. The four known forces of physics appear in an inseparable combination. Gravity, in particular, emerges as a residual effect of the electromagnetic force in this scenario, resulting in a deterministic toy universe driven by a single input parameter. An intrinsic, non-local, mechanism gives support to the rich results from quantum physics. The proposed model is developed using a constructive approach, leading to a universal cellular automaton, where information has an ontological character. This single-world solution is not an interpretation of Quantum Mechanics, but a deeper attempt to describe nature.

PACS numbers: 12.10.Kt

Keywords: cellular automaton, nonlocality, emerging gravity, unification, cardinality

1 Introduction

Wheeler [1] coined the aphorism 'it from bit'. With this, he meant that anything physical, any *it*, derives its existence from discrete binary choices, or *bits*. This gives support to the notion that information has an ontological nature. The concept implies that physics, particularly quantum physics, isn't really about reality, but just our best description of what we observe.

In this regard, cellular automata (CAs) are mathematical idealizations of physical systems in which space and time, an evolution parameter, are discrete. Their attractiveness comes from the notion that simple rules can lead to very complex behavior, tending to long and interesting evolutions.

*alexandre.com@yahoo.com

An alternative representation of the universe is developed in this work using the cellular automaton paradigm. The theme has been explored for a long time (see [2, 3, 4, 5, 6, 7, 8, 9, 10, 11] for example). However, these studies generally remain in the abstract realm or present very limited models. Here, a full **3+1** core specification is posited. Although it is a qualitative analysis for the time being, the model is amenable to immediate computational investigation.

Just classical logic and plain integer math, along with a hint of topology, are used in the dynamics, making up a constructive approach.

This initiative was driven by the difficulty of unifying the two pillars of physics, quantum mechanics and general relativity, in a supposed quantum gravity. Such difficulty, in the author's view, is due to the introduction of imaginary numbers both in Schrodinger's equation and in the spacetime metric.

It will be clear that foundationally this is not an interpretation of Quantum Mechanics, but a deeper attempt to describe nature.

Finally, the reader will be delighted when he sees that purely intuitive arguments are used, leaving out obscure concepts.

2 Space and time

2.1 The register tensor

Each cell contains the same amount of information characterized by a bit string formatted as shown in table 2.1.

This structure is called a *tensor*. The terms *tensor* and *cell* are equivalent.

2.2 The espacito

Let *SIDE* be an integer value. A three-dimensional grid formed by $SIDE^3$ cells is called an *espacito* (small space in portuguese). Espacitos are a kind of supercell. In general, cell information propagates to adjacent spacitos relative position plus one position in two steps. The final position will be

$$x' = x + (SIDE + 1) \text{ mod } SIDE.$$

These jumps are necessary for interaction detection when confronting the stable and draft lattices (see Section 2.3). The actual pattern is three-dimensional.

Notice that an espacito occupies one single spatial point in the model, that is, it is an abstract concept.

2.3 The host lattice

A couple of coexisting *3d* Euclidean *lattices* made of $SIDE^3$ espacitos each, closed on themselves as three-dimensional tori, represents an absolute inertial reference frame. All cells in all espacitos are synchronized by a common, unidirectional, Newtonian time $t \in \mathbb{N}$, that is, they pulsate in unison, exchanging information with its seven neighbors (six in dimensions *xyz* and one in dimension *u*, used for the massive transfer of information from one lattice to the other). This exchange is done alternately, so time homogeneity is recovered at each couple of five clock ticks, or one *pass* (see further details below), configuring a *cellular automaton*.

The distance between the cells along the three spatial axes corresponds to the fundamental quantity *X* in the real world, with

Name	Type	Symbol	Obs.
Physical properties			
charge	B6	ch	Bits q , w_1 , w_0 , c_2 , c_1 , c_0
momentum	V	\mathbf{p}	motion driver
spin	V	\mathbf{s}	rotation driver
affinity	U	a	Used to group bubbles into packets
Wavefront			
lifetime	U	t	bubble's lifetime
origin	V	\mathbf{o}	Also used in the detection of visits.
synchronism	U	syn	spherical propagation timing
sine gen.	S	u, v	v carries the sinusoidal phase pmf
frequency	U	f	Number of clumped bubbles for a given a
Footprint			
propensity	V	p_0	momentum propensity
direction	V	\mathbf{prone}	sought for direction
Superluminal variables			
flash	U	$flash$	Enforces an ontological collapse
pole	V	\mathbf{pole}	Mainly used in the inertia mechanism
target	U	target	affinity collapsing
Interaction control			
kind	U8	$kind$	Encodes an interaction between bubbles
noise	U	$noise$	Interaction test

Table 1: The tensor.

The formatted bit string, where U means unsigned integer, S signed integer, V three-dimensional integer vector and B a boolean. The convention for the values *false* and *true* for each charge bit is: q : +, -; w_1 : [O]rbis, [D]ark; w_0 : [R]ight, [L]eft; c_2 : [B]lue, anti[\bar{B}]lue; c_1 : [G]reen, anti[\bar{G}]reen; c_2 : [R]ed, anti[\bar{R}]ed.

$$X = \frac{\ell_P}{2^{59}},$$

where ℓ_P is the Planck length.

The definition of the fundamental quantity of time is postponed to the next section.

The single input parameter *SIDE* was estimated from the size of the observable universe in terms of X^1 as

$$\begin{aligned} SIDE &= 2^{order}, \\ &\approx 9.48 \times 10^{80}, \end{aligned}$$

where *order* = 269. The arithmetic uses *SIDE* as module. The number of spacitos is then $2 \times SIDE^3$.

The torus is the simplest topological structure capable of holding a 3D universe and charge quantization. The finitude of the universe is a necessary condition to reach this quantization (Smith [12]). Cosmological observations favor a multi-connected rather than a simply connected universe (see Lachieze-Rey and Luminet [13]).

3 Bubbles

There are two types of collective spread of cellular information. One is a greedy spread in the raw ticking of the clock called *flash*, shaped like a cubic wave front. The other follows a spherical pattern and will be considered below.

A *bubble* is an expanding spherical wavefront of information organized on the tensors. Whenever vector \mathbf{p} aligns with vector \mathbf{o} , such a perturbation is cast. The von Neumann neighborhood *Dirs* is explored in a Case, Rajan and Shende algorithm from [14]. There they show that the accommodation of the wavefront takes $2D$ clock passes (one light step, *LIGHT*), where D is the cube diagonal, so bubbles expand with the same speed c , increasing $|\mathbf{o}|$ in the modified tensors. In some cases, concurrent access to a common cell is attempted, being *per se* an undecidable problem. A simple algorithm is used to overcome this constraint and select a path. Incidentally, this direction index behaves like a random number of module 6, emerging as the basis for decision making in interactions. (see Chaitin [16] arguing the intrinsic randomness of mathematics). Only time *LIGHT* has physical meaning, the time tick is only for low-level sync. It should be clear now that the *espacito* concept was created to support the overlap of bubbles.

We can now state the following constraint to the fundamental quantities

$$\frac{X}{LIGHT} = c.$$

A bubble is eventually reissued from a point calculated on its surface by resetting the origin vector ($\mathbf{o} = \mathbf{0}$) and/or aligning vectors \mathbf{o} and \mathbf{p} when interacting with other bubbles.

Each bubble lives in a separate 3d-space L_i , $i = \{1...SIDE^3\}$ to avoid propagation issues. The cells having the same *xyz* coordinates inside all *espacitos* of a lattice form one such a separate space. As bubbles propagate, they move continuously from one *spacito* to another, as will be clear below.

¹The exact granularity of the universal fabric will be obtained when it is verified that the model generates the same number of particles and the same amount of energy present in the universe. This provisional value was estimated using eight times the Eddington number, thus taking into account electrons, antiparticles as well as dark matter.

Most properties in tensors spread to expanding cells (see table above), while property *neighbors* is immutable, since it is the '*wires*' that connect cells to form a lattice. In particular, vector \mathbf{p} is copied to the occupied cells. As soon as the bubble leaves that site, it abandons this variable by setting $f = 0$, which in turn decays in time until value zero, as given by Equation (1).

F bubbles with a common affinity $a_j = A$, $j = \{1...F\}$, have a frequency $f = F$, their *de Broglie frequency*. Two such bubbles with some or all opposite charges and having $\mathbf{o}_1 = \mathbf{o}_2$ (that is, superposing) may form a *pair* if they are set to a common affinity value. Two extra pairs breaking this rule are the neutrino and antineutrino pairs ν and $\bar{\nu}$, with color values N, N and \bar{N}, \bar{N} , and the up quark fragments u , with equal, non trivial, colors and positive charge.

The space is endowed with memory, which records the visit of the particles. This contributes to the appearance of self-interference, as in the case of the double slit experiment. This scheme was originally conceived by Sciarretta in [10]. The main idea is to leave a copy of the visitor's momentum on the lattice, which decays following the rule

$$p[t + 1] = p[t] \left(1 - \left(\frac{p_0}{t} \right)^2 \right), \quad (1)$$

being eventually passed to the evolving *bubble* (defined in Section 3), which in turn decays as

$$p[t + 1] = p[t] \left(1 - \frac{1}{2t} \right), \quad (2)$$

where p_0 is the momentum since the last transfer to the particle and t is the lifetime of the particle.

Taking all contributions into account in a typical tabletop experiment, after many repetitions of equally prepared particles (or an ensemble, for that matter), this value converges to the known QM probability densities involving typically a square root and a space-dependent sine function. Note that these expressions must be carefully adapted for integer or rational arithmetic. This emerging sine function should not be confused with that inherent in electromagnetism described below.

The bubbles also have an embedded sine phase (variables u and v in the tensor structure) coupled to f , achieved by recursive application of the laws

$$v = n^2 SIDE^2; u = u(v - t^2). \quad (3)$$

Variable v represents the pmf of the sine value at the point. Moreover, they leave a footprint of their passage every time they are re-emitted, consisting of the vector \mathbf{o} , vector \mathbf{p} and affinity a . This track paves the way for the occurrence of self-interference.

During wavefront accommodation the value of variable *noise* is updated using the selected direction index ($Dirs[index]$) and later compared with signal v , which indicates whether a bubble-bubble interaction is legal — this produces the observed harmonic behavior at the ensemble level. The update rule in this case is

$$noise = (noise << 3) \wedge dir.$$

Charges are represented by bits. The *electric* charge q is associated as ever with attraction/repulsion. The three *color* charges c_2, c_1, c_0 enforce the strong force structure. Color codes are $N : 000$, $R : 001$, $G : 010$, $\bar{B} : 011$, $B : 100$, $\bar{G} : 101$, $\bar{R} : 110$, $\bar{N} : 111$ for *neutral*, *red*, *green*, *antiblue*, *blue*, *antigreen*, *antired* and *antineutral*, respectively. Let signature be $sig = c_2 + c_1 + c_0$. A bubble is matter M if $sig < 2$, or antimatter \bar{M} , otherwise. Also, it is

neutral N , if $sig = 0$ or anti-neutral \bar{N} , if $sig = 3$. The definition $color = 2^2c_2 + 2^1c_1 + 2^0c_0$ will be used later. On the other hand, the *weak* charge with its two bits w_1 and w_0 , or *chirality*, is associated with congruence. Since the universe is closed, the addition of an extra bit is not an *ad hoc* solution, but actually guarantees this sense of congruence — It is like marking a couple of poles on an otherwise completely dull sphere.

Vector \mathbf{p} is related to motion direction, while vector \mathbf{s} to spatial rotation. I will refer to them as *momentum* and *spin* for convenience. The cell on the surface of the bubble pointed by vector \mathbf{p} from its own center is defined as its *pole*. The default propagation of vector \mathbf{p} is to follow the direction on the bubble pointed to by vector \mathbf{s} .

Finally, a loose association between the terms *bubble* and (unit of) *energy* is assumed, so energy is a conserved property, since bubbles are never created or destroyed. Vector \mathbf{p} of each bubble never changes, so momentum is a globally conserved quantity also.

4 Interactions

4.1 Overview

Interactions are evaluated at the last pass of the wavefront accommodation. Prior to interaction, updates need to be done. For instance, pairs are classified according to their charge content. Also, the sine signal and noise which have been updated during said accommodation and are now ready to be used.

There are four possible results for the two interacting bubbles: *i*) they cross each other just changing their phases; *ii*) both are re-emitted from the contact point, *cp* (annihilation); *iii*) they are re-emitted from their respective poles (cohesion); and *iv*) one is re-emitted from the *cp*, while the other from the parallel transported pole (inertia). Figure 1 depicts these cases.

In the first case, the interaction does not change momentum.

When a bubble is reissued after interacting with another bubble and, in addition, their poles coincide (case *c*) above), all bubbles with the same affinity are also reissued, characterizing an ontological *collapse*. The possibility of the collapsing parts having a space-like separation is supported by the fact that the phase of accommodation of the wavefront lasts $2D$ passes, sufficient for the information to propagate to all regions of the lattice. This flash stimulates the bubbles to reissue and reset their affinity property.

4.2 Basic cycle

The detailed algorithmic description of all interaction rules is contained in the Appendix. Below I will make a brief description of those rules.

One lattice is named *stable*, while the other is the *draft*. The basic loop consists of the following steps

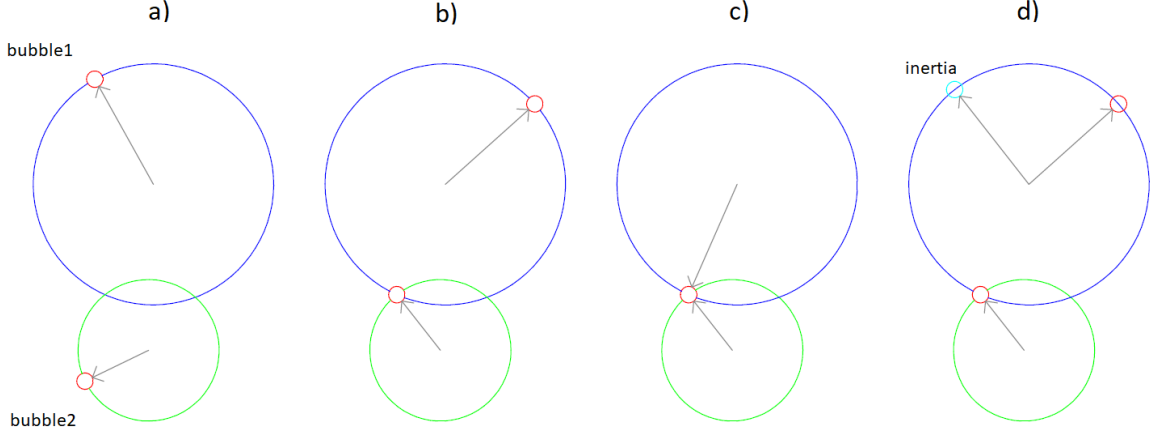
- **Copy**

Data is copied from the draft cells to the stable cells all at once.

- **Expand**

Two kinds of patterns propagate in this phase. The greedy expansion of the flash, and the spherical bubble. All bubbles expand synchronously one step in their respective *xyz* sub-lattice. In particular, non-conflict in cell access is guaranteed by algorithm (1).

Figure 1: General interaction types.



Four types of interactions can occur when the surfaces of the expanding bubbles cross each other: *a)* phase only; *b)* reissue of the two bubbles; *c)* collapse; *d)* parallel transport (inertia). The red ball is the momentum cell, the arrow is the momentum vector. The cyan ball is the parallel transported point. See Section 4.1 for details.

- **Pairing**

Content from the draft cells is moved to the adjacent separate space to allow comparison. That is, the draft and stable lattices are constantly being desaligned.

- **Update**

Before a *LIGHT* period is not complete, properties such as the sine phase, footprint tracking and pair classification are updated in the draft cells during this step. The sine phase generation relies on a recursive product (Equation 3).

- **Collision**

After a *LIGHT* period is completed the draft cell is compared with the stable cell for possible interactions (see Section 4.4 below). Besides using the sinusoidal phase, all possibilities of charge content and overlapping are considered to complete the interaction detection, being finally filtered by the selection rules of empirical evidences, being enforced or suppressed, guided by the weak charge.

A messenger is released at the end of the cycle, which, besides carrying the static forces, helps to enforce the collapse mechanism.

4.3 Initial state

Let S be the number of states resultant of the combination of all tensor values, all positions in the lattice and all bubbles L . There is an endless period loop $T_P < S$ with definite net Shannon entropy that corresponds to the real world.

Among the countless possibilities, I choose the following platonic solution for the initial state problem. All bubbles occupy the same espacito, no matter which, called the *singularity*.

Assume i is the index of the current cell in the singularity, $i = 0 \dots SIDE^3 - 1$. The weak charge is defined as $w_{0(i)} = i \bmod 2$ and $w_{1(i)} = i/2$. The electric charge is defined as $q_i = d_i \mathbf{xor} w_i$. Color is defined as $color = i \bmod 8$. Momentum \mathbf{p} starts with the sequence $[\mathbf{p}_0 = (+L, 0, 0); \mathbf{p}_1 = (0, +L, 0); \mathbf{p}_2 = (0, 0, +L); \mathbf{p}_3 = (-L, 0, 0); \mathbf{p}_4 = (0, -L, 0); \mathbf{p}_5 = (0, 0, -L)]$, which is recursively applied to the remaining indices. Of course this results in a perfectly symmetric configuration.

```

If  $\Delta x \neq 0$ 
     $\pm 1, 0, 0$ 
    If  $\Delta y \neq 0$ 
         $0, \pm 1, 0$ 
        ElseIf  $\Delta z \neq 0$ 
             $0, 0, \pm 1$ 
        Endif
    ElseIf  $\Delta y \neq 0$ 
         $0, \pm 1, 0$ 
        If  $\Delta z \neq 0$ 
             $0, 0, \pm 1$  synchronously
        Endif
    ElseIf  $\Delta z \neq 0$        $0, 0, \pm 1$  Else
         $\pm 1, \pm 1, \pm 1$ 
    Endif

```

where Δ is the difference between the position and the disturbance origin vectors and the signed numbers represent the increments per axis. This logic avoids attempt of simultaneous access to one cell.

Algorithm 1: Non-conflict algorithm.

Let $N \equiv SIDE^2$. For spin \mathbf{s} , we fill a particular plane (say, $z = 0$, so $i = 0 \dots N-1$) with the pattern above used for momentum, resulting in $[\mathbf{s}_0 = (+L, 0, 0); \mathbf{s}_1 = (0, +L, 0); \mathbf{s}_2 = (0, 0, +L); \mathbf{s}_3 = (-L, 0, 0); \mathbf{s}_4 = (0, -L, 0); \mathbf{s}_5 = (0, 0, -L)]$ for the head and $[\mathbf{s}_{N-4} = (+L, 0, 0); \mathbf{s}_{N-3} = (0, +L, 0); \mathbf{s}_{N-2} = (0, 0, +L); \mathbf{s}_{N-1} = (-L, 0, 0)]$ for the tail. We then replicate this pattern across all other planes. The perfect symmetry found in the first case gives room to a privileged spin direction, inducing one topologically trapped magnetic monopole, per sector, in line with Smith [12]. To complete, all bubbles start with $f_i = 1$, $t = 0$, $\mathbf{o}_i = \mathbf{0}$, $\mathbf{pole}_i = \mathbf{0}$, $code_i = 0$, $a_i = i$ and $noise_i = i$.

4.4 The sinusoidal phase

Here I'm going to dig a little deeper into the update/interact phases above.

Every bubble carries a sinusoidal phase kept with the help of an adapted product formula for the sine function. Variables u and v in the tensor structure form a rational number, being updated as

$$u_{n+1} = u_n (n^2 K^2 - t^2); \quad v = n^2 K^2.$$

The *noise* variable has a uniform distribution so it is fair to compare it against the well known PDF of the form

$$p_Y(y) = \frac{1}{\pi \sqrt{1 - y^2}} \quad (4)$$

where y is the sine amplitude, to trigger the interaction.² To use this PDF, we need to integrate a $1/N$ slice into the y position and compare it with a sample of a uniform distribution. Clearly, we must get rid of the square root and obtain the equivalent probability

²This implies the emergence of an intrinsic value of α for the model, reflecting the electromagnetic coupling.

mass function $p_Y(y)$, with support $R_Y = \{-N/2, +N/2\}$, to apply in our model using the value *noise* normalized to N as the random sample. With this in mind, we apply the Taylor series of the distribution for Equation (4) leading to

$$p_Y(y) = \sum_{n=0}^{\infty} \frac{(2n+1) y^{2n}}{2^n}. \quad (5)$$

The mechanism we have just described helps to enforce the interactions of static forces, i.e., the Coulomb radial attraction/repulsion and the magnetic lateral kick.

The accumulated values of Equation (5) remain stored on the lattice as the bubbles propagate, being immediately available when needed.

The sine generator must be bumped f times during the accommodation step, so that at the end of the *LIGHT* pass, it has the correct phase. For this, variable f of the draft cell decrements to zero, which happens early in the long accommodation step lasting $2D$. After the light step is concluded, the original value of f is recovered from the stable cell.

4.5 Footprint tracking

Equations (1) and (2) help implement a kind of "quantum potential". They rely on the detection of previous visits from the region where the particle was prepared. In order to decide on such a visit, the following two conditions should be assessed at the same time:

$$\mathbf{o}_1 \bullet \mathbf{o}_2 \cong 0$$

and

$$|\mathbf{o}_2 - \mathbf{o}_1| \cong 0,$$

where \mathbf{o}_1 is the current bubble origin vector and \mathbf{o}_2 comes from the last visit.

If no visit has been detected (due to a static force operation, maybe), there will be an exchange of information between the particle fragment and the lattice.

4.6 Collapse

The destruction of a particle due to annihilation or light-matter interaction (option *c*) *collapse* above) is performed by the messenger, who orders them all to be reissued, but only in the annihilation case each affinity property a will be redefined to a new random value

$$a = a \mathbf{xor} \text{ noise}.$$

Bubbles are now raw material for the formation (or recreation) of new particles.

4.7 Parallel transport

The last generic interaction case to be considered (letter *d*) above) is inertia, acquired by a parallel transport mechanism. It happens between a bubble with $f_1 = 1$ and a bubble with $f_2 > 1$, both with the same affinity $a_1 = a_2$. Fragment f_2 is reissued from the contact point (*cp*), while f_1 will be reissued from a pole corresponding to a parallel transported *cp*. Vector **pole** will be used to find the remote reissue point with the help of the messenger.

$$\mathbf{pole}_1 = \mathbf{o}_2 - \mathbf{o}_1.$$

As the messenger progresses, it decreases the appropriate coordinate until eventually it obtains $\mathbf{pole}_1 = \mathbf{0}$. At this destination point, a reissue is triggered.

4.8 Charge combination

The w_1 weak charge bit separates the interactions into two groups: same-sector and inter-sector. In what follows, we will first briefly describe the same-sector ($w_{1(1)} = w_{1(2)}$) cases and later the intersector case.

4.8.1 Coulomb interaction

Coulomb attraction and repulsion is done in two steps. If two fermion bubbles from different particles interact like in option *b*) above, where $\mathbf{p}_1 \neq \mathbf{0}$, then $\mathbf{prone}_1 = \mathbf{o}_1 - \mathbf{o}_2$ or $\mathbf{prone}_1 = \mathbf{o}_2 - \mathbf{o}_1$ for repulsion or attraction, respectively. If, in the other scenario, $\mathbf{p}_1 \neq \mathbf{0}$ and \mathbf{p}_2 is a photon bubble and $\mathbf{prone}_1 \neq \mathbf{0}$ and additionally $\mathbf{prone}_1 \bullet \mathbf{p}_2 = 1$, that is, they are aligned, then $a_2 = a_1$. The photon has been recruited, becoming a propeller, with the same affinity of the fermion from now on.

4.8.2 Magnetic lateral kick

Calculation of the prone vector is now $\mathbf{dif} = \mathbf{o}_2 - \mathbf{o}_1$, $\mathbf{prone} = \pm \mathbf{dif} \times \mathbf{s}_2$. The sign depends if charges are equal or different.

4.8.3 Neutrino interactions

A neutrino fragment is a pair of bubbles with charge equal to 000000*B*, while an antineutrino fragment is a pair with both bubbles having charge equal to 111111*B*. (Anti)Neutrino fragments interact with (right)left handed stuff only.

4.8.4 Intersector interaction

We can behold possible combinations of color properties between pairs of bubbles in table 4.8.5. (Anti)*Protoquark* fragments have non trivial color and are marked as q or \bar{q} , (anti)neutrinos are ν and $\bar{\nu}$, gluons are g , and photons are γ . Since half the gluon fragments are right handed, they are shown as photons. The cells marked as blue, if having both protoquarks with electrical charge $q = +$, may form an *up quark* fragment.

An *up quark* (u) can be any of the $+R+R$, $+G+G$ or $+B+B$ fragments, while a *down quark* (d) can be any of the $-R$, $-G$ or $-B$ fragments. When we combine three of these *uud* quark fragments to form an *atom* fragment, we get an uncompensated color. The matter end of the gluon may fit this lack, but in return it generates an antimatter end, and so on... Naturally, it follows that the *electron* fragment must be $-N, -N, -N$.³

By imagining a set of such fragments as a simplified Orbis, say, containing 256 bubbles, we can saturate it (the opposite scenario would be a sea of 128 photons) with 12 atoms with 6 photons between them (the other 6 photons were changed to inertial propellers), as well as the pair of neutrinos, plus a leftover of 12 positive leptonic ions. An escape valve for this strong ionization at the beginning is to allow intersector interaction ($w_{1(1)} \neq w_{1(2)}$) in the program code to balance things out. The electrical charge of the two fragments satisfying the free ion condition will be exchanged. As a result, six new photons will be added to each sector.

4.8.5 Symmetry breaking after singularity

Since all bubbles are overlapping at the singularity, an additional interaction rule must be conjured to separate the bubbles: If the two bubbles have the same weak and color charges,

³This explains the fractional charge of quarks.

Figure 2: Color combination.

	N	R	G	B	B	G	R	N
	000	001	010	011	100	101	110	111
N 000	ν	$1q$	$1q$	$1\bar{q}$	$1q$	$1\bar{q}$	$1\bar{q}$	γ
R 001	$1q$	$2q$	$2q$	γ	$2q$	γ	γ	$1\bar{q}$
G 010	$1q$	$2q$	$2q$	g	$2q$	γ	γ	$1\bar{q}$
\bar{B} 011	$1\bar{q}$	g	g	$2\bar{q}$	g	$2\bar{q}$	$2\bar{q}$	$1\bar{q}$
B 100	$1q$	$2q$	$2q$	g	$2q$	γ	γ	$1\bar{q}$
\bar{G} 101	$1\bar{q}$	g	g	$2\bar{q}$	γ	$2\bar{q}$	$2\bar{q}$	$1\bar{q}$
\bar{R} 110	$1\bar{q}$	g	g	$2\bar{q}$	γ	$2\bar{q}$	$2\bar{q}$	$1\bar{q}$
\bar{N} 111	γ	$1\bar{q}$	$1\bar{q}$	$1\bar{q}$	$1\bar{q}$	$1\bar{q}$	$1\bar{q}$	$\bar{\nu}$

Possible combinations of colored fragments are shown, where (anti)protoquark fragments are marked q and \bar{q} , (anti)neutrinos are ν and $\bar{\nu}$, gluons are g , and photons are γ . Since half the gluon fragments are right handed, they are shown as photons. The cells marked as blue, if having electric charge $q = +$, may form an *up quark* fragment (see Section 4.8.4 for details).

they are re-emitted from their own poles.

We are done with the axiomatic part by now. In the remainder of the paper, the arguments give support to the choices made and compose a minimalist interpretation, carefully avoiding unnecessary speculation.

4.9 Evolution

The system evolves following this stationary sequence of states, the very, very long *physical cycle* of period T_P , or Poincaré cycle. As the singularity unfolds, most bubbles are annihilated, creating a vacuum, both in *Orbis*, the sector with $w_1 = 0$, or in the *Dark Sector*, with $w_1 = 1$. The remaining bubbles within each sector, segregate on islands of same size with positive or negative charges (charge quanta) due to the aforementioned monopole. Antimatter tends to 'migrate' to the Dark Sector due to inter sector interactions. Observe that masses from Orbis are not seen from the Dark Sector and vice versa. Clearly, the two sectors coexist spatially.

4.10 Packets

Each region of space contains a homogeneous distribution of electrically neutral superposing bubbles (zero-point energy, or a vacuum), basically photons and Z particles, but also the charged W bosons, whether completely or partially formed. Some of these bubbles are recruited via the messenger interaction above, being dynamically added to a local *packet* of same affinity bubbles. If the packet contains colored or neutral non-superposing bubbles, different pairs will be captured from the vacuum.

Meanwhile, other bubbles wander away from the packet (dissipation) such that, when combined with the inherent non-linearity of the underlying system, an equilibrium situation is reached including many *propellers* (see definition below, Section 4.11), configuring in this way the *rest mass* of the packet (the resultant direction of the propellers defines the velocity of the packet). In other words, we have self-organization and entanglement⁴. The non-equilibrium caused by nearby charges or masses can change the motion of the packet, which in turn changes the effective mass.

Packets possessing a quantized set of non-superposing bubbles will have a collective “fermionic” character, while the pairs overlaid with their aligned spins give a packet a “bosonic” character, distinguished by their charge balance. The exception are the neutrino fragments, formed as NN (ν) or $\bar{N}\bar{N}$ ($\bar{\nu}$) which form a special pair. Since neutrino fragments interact with left handed stuff only, one of its spin \mathbf{s} is inactive, and therefore neutrinos also present a “fermionic” character (the same for antineutrinos, *mutatis mutandi*). In other words, if the packet includes a (quantized) population of non-overlapping bubbles with identical charge q then we have a *fermion*. Rather, if the population is formed by equal net weak charge, we have a neutral massive *boson*, and if the pairs also have non trivial electric balance then we have a charged weak boson. If, diversely, color is involved and no overlapping, then we have a quark fragment. Finally, if these bubbles are combined in pairs with the same non-neutral color, then we have a gluon fragment. The exception to the rule is the photon, which is an expanding shell ‘packet’. The individual bubbles of a fermion follow a random walk like pattern, indeed, an asymmetric one under the influence of the propellers.

Ephemeral resonance states, including radial vibration modes⁵ (see Itô [18]) induced by the presence of neutrinos (which also help to conserve angular momentum), are also possible, adding to the formation of a mass spectrum.

The *Hofer effect* is the expected tendency for all spins of a packet to align radially either inward or outward (spin up/down), as predicted in Hofer [19]. In that work, there is an explanation of how magnetic effects emerge from symmetry breaking of this spherical pattern, thereby supporting the Stern-Gerlach experiment.

4.11 Long range energy exchange

A pure *photon* γ is a multipair ($f = 2n$, $n > 0$) where each pair has all its charges in opposition and same \mathbf{p} and \mathbf{s} vectors. A photon fragment having the same affinity of a packet is dubbed a *propeller*.

The long range behavior of photons is due to the absence of other bubbles with the same a value to allow an inertial interaction (not a propeller), interacting instead either as a light-matter scattering or a refraction process above, possibly in a distant place.

It may happen that two bubbles in a packet, when interacting as in Section 4.2 above, are re-emitted from the same point, that is, their poles coincide, so a pair with the characteristics above may be formed. These coincidences may repeat so that overlapping of many bubbles having a common a value become possible. If they move away stimulated by another bubble-bubble process, a photon is released — spontaneous emission being a special case for faint fields. Simply put, photons are normally released in an atomic electronic decay, shaped by the spherical harmonics originating from charge quantization. This is the main explanation for the ubiquitous presence of all sort of quanta. The primary quantized quantity, is truly the electric charge.

⁴A recent study supports this view (see Zhi-Bo Yang et al. [15]).

⁵Empirically in three generations of leptons and quarks.

Photons also come similarly from *bremssstrahlung*. In this scenario, the spectrum is continuous since it has nothing to do with spherical harmonics.

The messenger *versus* bubble interaction also implies that photons may lose components while traveling (photon aging) thereby supporting General Relativity observations and the CMB.

4.12 Weak decay

No true randomness is involved in the calculation of the interactions, so the model is in fact deterministic, but the ubiquitous presence of weak charges, combined with the effects of variable *noise*, imply in an apparently random decay of particles. The W^- and Z bosons, even if not completely formed (virtual particles?), are responsible for it.

4.13 Higher order particle structures

When reorganizing after a collapse, bosons and fermions can be formed in a variety of ways, spins can be flipped etc. With all these ingredients, bound states of the strong and electromagnetic forces are a natural consequence.

5 Results

A small program (see Ref. [23]) was created to check the combination of charges. A total of $68 \times 128 = 1536$ fragments, with their charges evenly distributed, were randomly combined in two million attempts. This operation was repeated a hundred times and the averages tabulated. The results can be seen on Table 2. The leftover, that is, the bubbles that could not form a pair or a single, show a highly ionized environment, that will be attenuated by intersector interactions.

This result resembles roughly 30 simple 'atoms', while the photon fragments could form a few multiple frequency photons (30 say). The unmatched quark fragments could form mesons or virtual quarks. All in all, it makes a lot of sense as a toy universe.

6 Discussion

A toy universe was built from scratch using a constructive approach with a reduced set of ingredients. G. 'tHooft [11] shows that systems of this type can be associated to a permutation operator and, therefore, can be mapped to a large Hilbert space. Thus, at least in principle, all the machinery of operator mechanics, in particular the Schrödinger equation, can be used to analyze them — the point here is that the system certainly presents a quantum behavior. In this context, Born's rule arises naturally without the need to be postulated, bringing with it quantum probabilities. As a consequence, violation of Bell-type inequalities are expected, induced by the intrinsic non-locality. Perhaps new analysis tools should be devised to explore the model in search of physics beyond the Standard Model.

The proposed model is non-local under the light time basis, although being strictly local on the ticking basis of the clock and therefore no signaling is possible at any classical limit.

An intuitive argument allows us to associate *SIDE* to the much sought for cardinality of natural numbers \aleph_0 (see Asperó and Schindler [20]) — the maximum number with physical meaning in the universe is just a simple function of *SIDE* ($SIDE^3$, perhaps?).

Table 2: Charge combination

Fragment	Type	Quantity	Obs.
Neutrino	Pair	40.5	
Gluon	Pair	127.6	
Up quark	Pair	31.7	
Photon	Pair	195.6	
Z boson	Pair	112.2	
W boson	Pair	1.2	
Electrons	Single	88.9	
Quarks	Single	258.7	
Leftover	Single	171.1	O[+]=40.9, O[-]=70.7, D[+]=9.5, D[-]=49.9
Total		1536.0	

A total of 1536 bubbles were randomly combined into two million attempts. This operation has been repeated a hundred times (See Section 5 for more details).

Last but not least, a gravity-like dynamics emerges as a residual effect of the electromagnetic force (see also Assis [21]) — the halos formed in the sea of messengers around the masses caused by the absorption of messengers (perceived as not shielding because of the relatively low number of messengers absorbed) causes an imbalance in the distribution of momentum, resulting in an always attractive phenomenon, or gravity.

The existence of two worlds (Orbis and Dark sector) slightly coupled by gravity and the weak force give rise to Moiré patterns, which might in turn explain the observed cosmological constant and the Higgs field (see [22]). Gravity also seems to be responsible by the spin statistics connection.

We are, therefore, facing an ontological⁶ economic framework on a subplankian scale. This single-world solution promises to be robust at high momentum transmitted to the package / particle (e.g. partons) as in nature. Observable reality is approximate and emergent. The huge number of bubbles forming the particles — indeed a mini universe each — gives material support to superpositions and qubits. These rules can be refined, given sufficient computational power and programming support⁷, and evaluated if they in fact allow building a predictive, *bona fide*, theory.

Funding and competing interests

The author declares that no funds, grants, or other support were received during the preparation of this manuscript.

The author has no relevant financial or non-financial interests to disclose.

⁶Ontology is actually an always receding rule marking the frontier of the unfathomable.

⁷An implementation under development is accessible in Reference [23]

References

- [1] Wheeler, J.A. *Information, physics, quantum: the search for links*, In Complexity, entropy and the physics of information (ed. W. Zurek). Reading, MA: Addison-Wesley (1990).
- [2] Zuse, K. *Rechnender raum*, Elektronische Datenverarbeitung, **8**, 336-344 (1967).
- [3] Feynman, R. *The character of the physical law*, MIT, ISBN 0 262 56003 8 (1967).
- [4] Gardner, M. *The fantastic combinations of John Conway's new solitaire game "life"*, Sci. Am. **223**, 120-123 (1970).
- [5] Margolus N. *Universal cellular automata based on the collisions of soft spheres*, Adamatzky A. (eds) Collision-Based Computing. Springer, London (2002).
- [6] Wolfram, S. *A new kind of science*, Wolfram Media, {23-60}, 112, and 865-866 (2002).
- [7] Fredkin E. *An introduction to digital philosophy*. Int. J. Theor. Phys. **42** (2): 189–247 (2003).
- [8] Elze, H.T. *Action principle for cellular automata and the linearity of quantum mechanics*, Phys. Rev. A **89** 012111 (2014).
- [9] H.-T Elze *Qubit exchange interactions from permutations of classical bits*, Int. J. Quant. Info., **17** (08), 1941003 (2019).
- [10] Sciarretta, A. *A local-realistic model of quantum mechanics based on a discrete space-time*, Found. Phys. **48** (1), 60–91, (2018).
- [11] 't Hooft, G. *The cellular automaton interpretation of quantum mechanics*, In: van Beijeren, H., et al. (eds.) Fundamental Theories of Physics. Springer, Berlin (2016).
- [12] Smith, W.D. *Charge quantization, the topology and 3-dimensionality of the universe, and abolishing monopoles*, <https://www.semanticscholar.org> (2000).
- [13] Lachieze-Rey M., and J.P. Luminet *Cosmic topology*, Phys. Rept. **254** (1995).
- [14] Case, J., D. Rajan, and A. Shende *Spherical wave front generation in lattice computers*, Int. J. Comput. Inf. (1994).
- [15] Yang, Zhi-Bo et al. *Entanglement emerges from dissipation-structured quantum self-organization*, arXiv:2109.12315v1 (2021).
- [16] G. J. Chaitin, *Algorithmic Information Theory*, in IBM Journal of Research and Development, **21**, no. 4, pp. 350-359 (1977).
- [17] C. Turner, *Recursive discrete time sinusoidal oscillators*, IEEE Signal Processing Mag, p103-111 (2003).
- [18] Itô, D. *Cohesive force of electron and Nambu's mass formula*, Prog. Theor. Phys. **47**, no. 3 (1972).
- [19] Hofer, W.A. *Elements of physics for the 21st century*, J. Phys.: Conf. Ser. **504**, 012014 (2014).

- [20] Asperó, D., R. Schindler *Martin's Maximum implies Woodin's axiom (*)*. Annals of Mathematics, **193** (3), 793-835 (2021).
- [21] Assis, A.K.T. *Deriving gravitation from electromagnetism*, Can. J. Phys. **70**, 330-340 (1992).
- [22] Parhizkar A. V. Galitsky *Moiré gravity and cosmology*, Arxiv: 2204.06574v1 (2022).
- [23] Furtado Neto, A. *It from bit - a concrete attempt*, GitHub repository, <https://github.com/automaton3d/automaton> (2021).

Appendix: source code

Here are gathered all the rules of information exchange between cells of the cellular automaton. They are the essence of this work, intended to possess an **axiomatic** character.

I have used a C-like notation instead of traditional algorithm convention for clearness and fast implementation. No subroutines were used, nor were any optimizations attempted to keep the code as general as possible. The unique parameter *ORDER* comes from Section 2.

```
1  ////////////////////////////////////////////
2  //                                     //
3  // — Universe's software — //
4  //                                     //
5  ////////////////////////////////////////////
6
7  // Lattice symbols
8
9  #define ORDER      269 (see Section 2.3)
10 #define SIDE       (1<<ORDER)
11 #define SIDE2      (SIDE*SIDE)
12 #define SIDE3      (SIDE*SIDE2)
13 #define SIDE4      (SIDE*SIDE3)
14 #define SIDE5      (SIDE*SIDE4)
15 #define SIDE6      (SIDE*SIDE5)
16 #define MASK       (SIDE-1)
17 #define DIAG       (2*MASK)
18 #define LIGHT      (2*DIAG)
19 #define LIGHT2     (LIGHT*LIGHT)
20 #define S          (SIDE/2)
21 #define TOL         (SIDE/1024) // Provisional value
22 #define LIMIT      (3*(SIDE2-2*SIDE+1))
23
24 // Particle symbols (ored in code)
25
26 #define PHOTON      0x0001
27 #define GLUON       0x0002
28 #define NEUTRINO    0x0004
29 #define ZB          0x0008
30 #define WB          0x0010
31 #define UP          0x0020
32 #define DOWN        0x0040
33
34 #define FERMION     0x0080
35
36 // Charge masks
37
38 #define C_MASK      0x07
39 #define Q_MASK      0x08
40 #define W0_MASK     0x10
41 #define W1_MASK     0x20
42
```

```

43 // Macros (helps readability)
44
45 #define ISNULL(v)      (v[0]==0 && v[1]==0 && v[2]==0)
46 #define ISEQUAL(v,u)  (v[0]==u[0] && v[1]==u[1] && v[2]==u[2])
47 #define RESET(v)      {v[0]=0;v[1]=0;v[2]=0;}
48 #define COPY(u,v)      {u[0]=v[0];u[1]=v[1];u[2]=v[2];}
49 #define MOD2(v)         (v[0]*v[0]+v[1]*v[1]+v[2]*v[2])
50 #define NEG(v)          {v[0]=-v[0];v[1]=-v[1];v[2]=-v[2];}
51 #define DOT(u,v)        (u[0]*v[0]+u[1]*v[1]+u[2]*v[2])
52 #define GETC(u)         (u->ch & C_MASK)
53 #define GETW1(u)        ((u->ch & W1_MASK) == W1_MASK)
54 #define GETW0(u)        ((u->ch & W0_MASK) == W0_MASK)
55 #define GETQ(u)         ((u->ch & Q_MASK) == Q_MASK)
56 #define SUB(v,v1,v2)    {v[0]=v2[0]-v1[0];v[1]=v2[1]-v1[1];
57                          v[2]=v2[2]-v1[2];}
58 #define CROSS(a,b,c)    {a[0]=b[1]*c[2]-b[2]*c[1];
59                          a[1]=b[2]*c[0]-b[0]*c[2];
60                          a[2]=b[0]*c[1]-b[1]*c[0];}
61 #define ISMATTER(u)     (GETC(u)>2&&GETC(u)!=4)
62
63 // Handedness
64
65 #define RIGHT  0
66 #define LEFT   1
67
68 // Tensor structure
69 // Practical types are used rather than conceptual types
70
71 typedef struct Tensor
72 {
73     // Physical properties
74
75     unsigned char ch; // charge
76     int p[3], s[3];   // momentum, spin
77     unsigned a;       // affinity
78
79     // Wavefront
80
81     unsigned t;        // lifetime
82     unsigned tt;       // lifetime
83     int o[3];          // origin
84     unsigned syn;      // synchronism
85     int u, v;          // Euler
86     unsigned f;        // frequency
87
88     // Footprint
89
90     int p0[3];         // momentum propensity
91     int prone[3];      // sought for direction
92

```

```

93  // Superluminal variables
94
95  unsigned char flash; // flash
96  unsigned pole[3];    // pole
97  unsigned target;     // affinity collapsing
98
99  // Pointers
100
101  unsigned offset;      // offset inside espacito (constant)
102  struct Tensor *wires[6]; // wires to other cells (constant);
103
104  // Interaction data
105
106  unsigned char kind;   // kind of fragment
107  unsigned noise;      // pseudorandom seed
108
109 } Tensor;
110
111
112 // Current pointers
113
114 Tensor *stbl, *draft;
115
116 /**
117  * Unique function to be executed simultaneously by all cells.
118  * Conceptually, each of the four steps below takes the same
119  * amount of Newtonian time to execute, or one clock tick,
120  * totalizing one pass.
121  */
122 void dynamics()
123 {
124  //////////////////////////////////////////
125  // Step 1: COPY (see Section 4.2) //
126  //////////////////////////////////////////
127
128  stbl->ch      = draft->ch;
129  stbl->a       = draft->a;
130  stbl->t       = draft->t;
131  stbl->tt      = draft->tt;
132  stbl->syn     = draft->syn;
133  stbl->u       = draft->u;
134  stbl->v       = draft->v;
135  stbl->f       = draft->f;
136  stbl->flash   = draft->flash;
137  stbl->kind    = draft->kind;
138  stbl->noise   = draft->noise;
139  stbl->target  = draft->target;
140  stbl->offset  = draft->offset;
141  COPY(stbl->p, draft->p);
142  COPY(stbl->s, draft->s);

```

```

143 COPY(stbl->o, draft->o);
144 COPY(stbl->pole, draft->pole);
145 COPY(stbl->p0, draft->p0);
146 COPY(stbl->prone, draft->prone);
147 pthread_mutex_unlock(&mutex);
148
149 //////////////////////////////////////
150 // Step 2: FLASH (see Section 4.2) //
151 //////////////////////////////////////
152
153 // Flash greedy expansion
154
155 if (stbl->flash == 0)
156     return;
157
158 // Flash decay
159
160 draft->flash = stbl->flash - 1;
161
162 // Test if pole was found (momentum must be present)
163
164 int f = stbl->f;
165 if (!ISNULL(stbl->p) && ISNULL(stbl->pole) && stbl->f > 0 &&
166     ISNULL(stbl->o))
167 {
168     // Pole found: create new seed
169
170     COPY(draft->p, stbl->p);
171     draft->f = stbl->f;
172     RESET(draft->pole);
173     RESET(draft->o);
174     draft->target = stbl->a; // non trivial target
175     f = 0;                // warn that pole was found
176 }
177
178 // Collapse forces all affine bubbles to reissue
179
180 if (stbl->a == stbl->target)
181 {
182     // Reissue
183
184     draft->f = stbl->f;
185     RESET(draft->pole);
186     RESET(draft->o);
187     draft->target = stbl->a; // non trivial target
188 }
189
190 // Explore von Neumann directions
191
192 int org[3];

```

```

193 Tensor* neighbor;
194 for (int dir = 0; dir < 6; dir++)
195 {
196     // Wrapping condition
197
198     COPY(org, stbl->o);
199     switch(dir)
200     {
201         case 0:
202             if (org[0] == SIDE - 1 || org[0] < 0)
203                 continue;
204             org[0]++;
205             break;
206         case 1:
207             if (org[0] == -SIDE + 1 || org[0] > 0)
208                 continue;
209             org[0]--;
210             break;
211         case 2:
212             if (org[1] == SIDE - 1 || org[1] < 0)
213                 continue;
214             org[1]++;
215             break;
216         case 3:
217             if (org[1] == -SIDE + 1 || org[1] > 0)
218                 continue;
219             org[1]--;
220             break;
221         case 4:
222             if (org[2] == SIDE - 1 || org[2] < 0)
223                 continue;
224             org[2]++;
225             break;
226         case 5:
227             if (org[2] == -SIDE + 1 || org[2] > 0)
228                 continue;
229             org[2]--;
230             break;
231     }
232     neighbor = draft->wires[dir];
233
234     // Flash is short-lived
235
236     neighbor->flash = 1;
237
238     // Propagate sought affinity for collapse
239
240     neighbor->target = stbl->target;
241
242     // Propagate particle presence

```

```

243
244     neighbor->f = f;
245     COPY(neighbor->o, org);
246
247     // Propagate momentum
248
249     if (f > 0)
250     {
251         COPY(neighbor->p, stbl->p);
252     }
253     else
254     {
255         RESET(neighbor->p);
256     }
257
258     if (!ISNULL(stbl->pole) && stbl->f > 0)
259     {
260         // Shrink pole as flash approaches re-emission point
261
262         neighbor->pole[0] = stbl->pole[0] - (dir < 2);
263         neighbor->pole[1] = stbl->pole[1] - (dir %2 == 1);
264         neighbor->pole[2] = stbl->pole[2] - (dir > 3);
265     }
266 }
267
268 ////////////////////////////////////
269 // Step 3: EXPAND (see Section 4.2) //
270 ////////////////////////////////////
271
272 // Test wrapping
273
274 if(stbl->o[0] * stbl->o[0] + stbl->o[1] *
275     stbl->o[1] + stbl->o[2] * stbl->o[2] == LIMIT)
276 {
277     // Brand new
278
279     draft->t = 0;
280     draft->syn = 0;
281     RESET(draft->o);
282     RESET(draft->pole);
283     draft->p[0] = 1;
284 }
285
286 // Increment lifetime
287
288 draft->t++;
289
290 // Is wavefront synchronized? (see Ref. [15])
291
292 if (draft->t * draft->t <= stbl->syn)

```

```

293     return;
294     if(stbl->f == 0)
295     return;
296
297     // Explore von Neumann directions
298
299     int org[3];
300     Tensor* neighbor;
301     int dotMax = 0;
302     Tensor *dst = NULL;
303     for (int dir = 0; dir < 6; dir++)
304     {
305         // Initialize new origin vector
306
307         COPY(org, stbl->o);
308
309         // Test if branch is legal
310
311         neighbor = draft->wires[dir];
312         if (neighbor->f > 0)
313             continue;
314         switch(dir)
315         {
316             case 0:
317                 if (org[0] == SIDE - 1 || org[0] < 0)
318                     continue;
319                 org[0]++;
320                 break;
321             case 1:
322                 if (org[0] == -SIDE + 1 || org[0] > 0)
323                     continue;
324                 org[0]--;
325                 break;
326             case 2:
327                 if (org[1] == SIDE - 1 || org[1] < 0)
328                     continue;
329                 org[1]++;
330                 break;
331             case 3:
332                 if (org[1] == -SIDE + 1 || org[1] > 0)
333                     continue;
334                 org[1]--;
335                 break;
336             case 4:
337                 if (org[2] == SIDE - 1 || org[2] < 0)
338                     continue;
339                 org[2]++;
340                 break;
341             case 5:
342                 if (org[2] == -SIDE + 1 || org[2] > 0)

```

```

343         continue;
344         org[2]--;
345         break;
346     }
347
348     // Check if is seed cell
349
350     int dot = org[0] * stbl->s[0] + org[1] * stbl->s[1] + org[2]
        * stbl->s[2];
351     if(dot > dotMax)
352     {
353         dotMax = dot;
354         dst = neighbor;
355     }
356
357     // Propagate all information
358
359     neighbor->t      = draft->t; // copy the updated time
360     neighbor->a      = stbl->a;
361     neighbor->ch     = stbl->ch;
362     neighbor->tt     = stbl->tt;
363     neighbor->u      = stbl->u;
364     neighbor->v      = stbl->v;
365     neighbor->f      = stbl->f;
366     neighbor->kind   = stbl->kind;
367     neighbor->target = stbl->target;
368     neighbor->offset = stbl->offset;
369     COPY(neighbor->pole, stbl->pole);
370     COPY(neighbor->o, org);
371     COPY(neighbor->s, stbl->s);
372     COPY(neighbor->p0, stbl->p0);
373     COPY(neighbor->prone, stbl->prone);
374     //COPY(neighbor->p, stbl->p);
375     RESET(neighbor->p); //????
376
377     // Update noise
378
379     neighbor->noise = (neighbor->noise << 3) ^ dir;
380
381     // Set timing for spherical pattern (Section 3)
382
383     neighbor->syn = LIGHT2 * MOD2(org);
384 }
385
386 // Propagate momentum
387
388 if (!ISNULL(stbl->p))
389     COPY(dst->p, stbl->p);
390
391 // Abandon the origin cell.

```



```

392 // Tracking information remains.
393
394 draft->f = 0;
395
396 //////////////////////////////////////
397 // Step 4: UPDATE (see Section 4.2) //
398 //////////////////////////////////////
399
400 // Is there a bubble in this cell?
401
402 if (stbl->f)
403 {
404     // Bump sine generator
405     // (Euler formula for sine)
406     int v = stbl->t * stbl->t * SIDE2;
407     draft->u = stbl->u * (v - stbl->t * stbl->t);
408     draft->f--; // original value is preserved in the stable
409                 cell
410
411     // Bubble track decay (Equation 1)
412
413     if (draft->f == 0 && !ISNULL(draft->p))
414     {
415         // Vector p shrinks
416
417         draft->p[0] -= draft->p[0] / (2 * draft->t);
418         draft->p[1] -= draft->p[1] / (2 * draft->t);
419         draft->p[2] -= draft->p[2] / (2 * draft->t);
420     }
421
422     // Update lattice decay
423
424     if (stbl->f == 0 && stbl->tt > 0)
425     {
426         // Lattice track decay (Equation 2)
427
428         draft->p[0] -= draft->p[0] * draft->p0[0] /
429                     (draft->tt * draft->tt);
430         draft->p[1] -= draft->p[1] * draft->p0[1] /
431                     (draft->tt * draft->tt);
432         draft->p[2] -= draft->p[2] * draft->p0[2] /
433                     (draft->tt * draft->tt);
434         draft->tt--;
435     }
436
437     // Find new address for interaction
438
439     int x = ((stbl->offset & MASK) +
440             stbl->o[0]) % SIDE ;

```

```

441  int y = (((stbl->offset >> ORDER) & MASK) +
442          stbl->o[1]) % SIDE;
443  int z = ((stbl->offset >> ORDER >> ORDER) +
444          stbl->o[2]) % SIDE;
445  Tensor *dst = stbl + x + y * SIDE +
446              z * SIDE2;
447
448  // Superposing bubbles?
449
450  if (ISEQUAL(stbl->o, dst->o))
451  {
452      // Clump bubbles together to form particle pair fragments
453
454      if (stbl->a != dst->a && stbl->f == 1 && dst->f == 1)
455      {
456          if ((GETC(stbl)^GETC(dst)) == C_MASK &&
457              GETW0(stbl) == !GETW0(dst) &&
458              GETQ(stbl) == !GETQ(dst))
459              draft->kind = PHOTON;
460          else if (GETC(stbl) == GETC(dst) &&
461                  GETW0(stbl) == !GETW0(dst) &&
462                  GETQ(stbl) == !GETQ(dst))
463              draft->kind = GLUON;
464          else if (GETC(stbl) == ~GETC(dst) &&
465                  GETW0(stbl) == GETW0(dst) &&
466                  GETQ(stbl) == !GETQ(dst))
467              draft->kind = ZB;
468          else if (GETC(stbl) == ~GETC(dst) &&
469                  GETW0(stbl) == GETW0(dst) &&
470                  GETQ(stbl) == GETQ(dst))
471              draft->kind = WB;
472          else if (GETC(stbl) == GETC(dst) &&
473                  GETC(stbl) != 0 && GETC(stbl) != 7 &&
474                  GETQ(stbl) == GETQ(dst) && GETQ(stbl) == 0)
475              draft->kind = UP;
476          else if (stbl->ch == dst->ch &&
477                  (stbl->ch == 0 || stbl->ch == 7))
478              draft->kind = NEUTRINO;
479          else
480              return;
481
482      // Calculate the common value for f and a
483
484      draft->a ^= stbl->a;
485      draft->f++;
486
487      // A pair is actually formed, since the other
488      // cell performs this same action in turn.
489  }
490

```

```

491 // Combine pairs to form multi pairs
492
493 else if (stbl->f % 2 == 0 && dst->f %2 == 0 &&
494 stbl->a != dst->a)
495 {
496 draft->f += stbl->f;
497 draft->a ^= stbl->a;
498
499 // Things fit at the end as above for
500 // the first pair
501 }
502 }
503
504 //////////////////////////////////////
505 // Step 5: INTERACT (see Section 4.2) //
506 //////////////////////////////////////
507
508 // Not the last pass of wavefront?
509
510 if (draft->t % LIGHT != 0)
511 return;
512
513 // No interaction possible?
514
515 if (!draft->kind)
516 return;
517
518 // Update the sine signal pmf (see Section
519
520 int v = stbl->t * stbl->t * SIDE2;
521 draft->v += ((2 * stbl->t + 1) * v * v) >> 1;
522
523 // Figure out the expected kind of interaction
524
525 int code = (stbl->a == draft->a) |
526 (!ISNULL(stbl->p) << 1) |
527 (!ISNULL(draft->p) << 2);
528
529 // Initialize collapse target
530
531 unsigned target = 0;
532
533 // Play pseudo dices to decide bubble mixing
534
535 if (stbl->noise < draft->v)
536 {
537 // Test for same or different sectors
538
539 if (GETW1(stbl) == GETW1(draft))
540 {

```

```

541 // Same-sector
542
543 switch(code)
544 {
545     case 1: // Soft interactions
546
547         if (stbl->a == draft->a)
548         {
549             // Self interference
550
551             int sub[3];
552             SUB(sub, draft->o, stbl->o);
553             if (DOT(stbl->o, draft->o) == 0 &&
554                 abs(MOD2(sub)) < TOL)
555             {
556                 // Sciarreta goes here
557             }
558         }
559     else
560     {
561         // Phase only interaction
562     }
563     break;
564
565     case 2: // Rendez vous (different particles)
566     case 4: // Rendez vous (different particles)
567
568         // Virtual photon capture
569         // (prone1 != 0, photon2, p2 aligned with prone1)
570         // This serves both to electrical or magnetic
571         // interaction
572
573         if (!ISNULL(stbl->prone) && draft->kind == PHOTON &&
574             DOT(stbl->prone, draft->p) == 0)
575         {
576             // Recruit it
577
578             draft->a = stbl->a;
579             RESET(draft->prone);
580         }
581
582         // fermion+ x fermion+ (partial scattering)
583         // fermion- x fermion- (partial scattering)
584         // repulsion
585
586         else if (stbl->f == 1 && draft->f == 1 &&
587                 GETQ(stbl) == GETQ(draft) && !ISNULL(stbl->p))
588         {
589             if (stbl->noise % 2 == 0)
590             {

```

```

591         // Electric radial force
592
593         SUB(draft->prone , draft->o , stbl->o);
594     }
595     else
596     {
597         // Magnetic lateral kick
598
599         int dif[3];
600         SUB(dif , draft->o , stbl->o);
601         CROSS(draft->prone , dif , draft->s);
602     }
603 }
604
605 // fermion+ x fermion- (partial scattering)
606 // fermion- x fermion+ (partial scattering)
607 // attraction
608
609 else if (stbl->f == 1 && draft->f == 1 &&
610         GETQ(stbl) != GETQ(draft) &&
611         !ISNULL(stbl->p))
612 {
613     if (stbl->noise % 2 == 0)
614     {
615         // Electric radial force
616
617         SUB(draft->prone , stbl->o , draft->o);
618     }
619     else
620     {
621         // Magnetic lateral kick
622
623         int dif[3];
624         SUB(dif , draft->o , stbl->o);
625         CROSS(draft->prone , dif , draft->s);
626         NEG(draft->prone);
627     }
628 }
629
630 // fermion- x boson (local light-matter)
631 // fermion+ x boson (local light-matter)
632
633 else if (stbl->f == 1 && draft->f > 1)
634 {
635     // boson x fermion
636
637     switch(draft->kind)
638     {
639         case PHOTON:
640

```

```

641          // Reemit from cp
642
643          RESET( draft->pole );
644          RESET( draft->o );
645          break;
646
647      case ZB:
648      case WB:
649          if (ISMATTER( stbl ))
650          {
651              if (GETW0( stbl ) == LEFT)
652              {
653                  if ((ISMATTER( draft ) &&
654                      GETW0( draft ) == LEFT) ||
655                      (!ISMATTER( draft ) &&
656                      GETW0( draft ) == RIGHT))
657                  {
658                      // Reemit from cp
659
660                      RESET( draft->pole );
661                      RESET( draft->o );
662                  }
663              }
664          }
665          else
666          {
667              if (GETW0( stbl ) == RIGHT)
668              {
669                  if ((ISMATTER( draft ) &&
670                      GETW0( draft ) == LEFT) ||
671                      (!ISMATTER( draft ) &&
672                      GETW0( draft ) == RIGHT))
673                  {
674                      // Reemit from cp
675
676                      RESET( draft->pole );
677                      RESET( draft->o );
678                  }
679              }
680          }
681          break;
682      }
683  }
684  else if (stbl->f > 1 && draft->f == 1)
685  {
686      // fermion x boson
687
688      switch( stbl->kind )
689      {
690          case PHOTON:

```

```

691
692         // Reemit from cp
693
694         RESET( draft->pole );
695         RESET( draft->o );
696         break;
697
698     case ZB:
699     case WB:
700         if (ISMATTER( stbl ))
701         {
702             if (GETW0( stbl ) == LEFT)
703             {
704                 if ((ISMATTER( draft ) &&
705                     GETW0( draft ) == LEFT) ||
706                     (!ISMATTER( draft ) &&
707                     GETW0( draft ) == RIGHT))
708                 {
709                     // Reemit from cp
710
711                     RESET( draft->pole );
712                     RESET( draft->o );
713                 }
714             }
715         }
716         else
717         {
718             if (GETW0( stbl ) == RIGHT)
719             {
720                 if ((ISMATTER( draft ) &&
721                     GETW0( draft ) == LEFT) ||
722                     (!ISMATTER( draft ) &&
723                     GETW0( draft ) == RIGHT))
724                 {
725                     // Reemit from cp
726
727                     RESET( draft->pole );
728                     RESET( draft->o );
729                 }
730             }
731         }
732         break;
733     }
734 }
735
736 // boson x boson
737
738 else if (stbl->f > 1 && draft->f > 1)
739 {
740     // Exchange charges and spins

```

```

741
742     COPY(draft->s , stbl->s);
743     draft->ch |= (stbl->ch & C_MASK);
744     draft->ch |= (stbl->ch & W0_MASK);
745
746     // Reemit from cp
747
748     RESET(draft->pole);
749     RESET(draft->o);
750 }
751 break;
752
753 case 3:    // Inertia
754
755     // Calculate parallel transported pole
756
757     SUB(draft->pole , stbl->o , draft->o);
758     RESET(draft->o);
759     break;
760
761 case 5:    // Inertia
762
763     // Counterpart in inertia is trivial
764
765     RESET(draft->pole); // cp
766     RESET(draft->o);
767     break;
768
769 case 6:    // Collapse
770
771     // fermion+ x fermion- (annihilation) ?
772     // fermion- x fermion+ (annihilation) ?
773
774     if ((stbl->ch & 0x1f) == ~(draft->ch & 0x1f) &&
775         stbl->kind == FERMION)
776     {
777         // Destroy particles , setting each 'a' property
778         // to a new, pseudorandom value
779
780         draft->a ^= stbl->noise;
781         draft->target = draft->a;
782         target = draft->target; // copy for flash
783         draft->kind = 0; // undo pairing
784     }
785     else if (stbl->kind == FERMION &&
786             draft->kind != FERMION)
787     {
788         // fermion- x boson (light-matter)
789         // fermion+ x boson (light-matter)
790

```



```

791         draft->a ^= stbl->noise;
792         draft->target = draft->a;
793         target = draft->target;    // copy for flash
794         draft->kind = 0;           // undo pairing
795     }
796     else if (stbl->kind != FERMION &&
797             draft->kind == FERMION)
798     {
799         // fermion- x boson (light-matter)
800         // fermion+ x boson (light-matter)
801
802         draft->a ^= stbl->noise;
803         draft->target = draft->a;
804         target = draft->target;    // copy for flash
805         draft->kind = 0;           // undo pairing
806     }
807     else
808     {
809         // fermion+ x fermion+ (scattering)
810         // fermion- x fermion- (scattering)
811
812         draft->target = draft->a;
813         target = draft->target;    // copy for flash
814     }
815
816     // All options in this case reissue from cp
817
818     RESET(draft->pole);
819     RESET(draft->o);
820     draft->target = stbl->a;    // non trivial target
821     break;
822
823     case 7:    // Internal collision
824
825         // Reissue from cp
826
827         RESET(draft->pole);
828         RESET(draft->o);
829         break;
830     }
831 }
832
833 // Intersector interaction
834 // (see Section 4.8.3)
835
836 else if ((code >> 1) == 3)
837 {
838     // Exchange electrical charges if ions
839
840     if (stbl->ch == 0x03 && draft->ch == 0x28)

```

```

841         draft->ch ^= Q_MASK;
842
843         // Reissue
844
845         RESET(draft->pole);
846         RESET(draft->o);
847     }
848 }
849
850 // Release a flash
851
852 draft->flash = 1;           // this low value vanishes quickly
853 draft->target = target;    // collapse target
854 }

```