Is Time a fundamental Category?

Suggesting Time as based on more fundamental Categories proposing a discrete Model on smallest Scales

Peter A. M. Möllers*

Delbrück, Germany

February 20, 2025 (Version 1.5)†

Abstract

Is *Time* a fundamental category and, if so, on the smallest thinkable scales? Maybe not. To test this conjecture this paper proposes a simple discrete model, the *Klick Model*. It bases *Time* on the correlation of two other categories, abstract definitions of *motion* and *state implementation* - an Informatics based approach to a problem in physics.[1]

The original motivation to develop the model was to understand the biological/atomic *ageing* of the twins of the so-called Twin-Paradox, seen as a *benchmark* combining time and space of smallest and largest scales. Without loss of generality, let us assume that the twins can be represented by two isolated (free) *abstract particles* in steady motion.

A model is only a model. It has to make falsifiable predictions for the real world. Therefore, I will show that for the Twin-Paradox it leads to the known result from Special Relativity Theory. For smallest scales the task is much more difficult. Only a *qualitative* test based on *structural similarities* can be offered here, defining a discrete logic for ground states, excited states and their correlations in the framework of the model, trying to map the abstract model to Einstein's discrete interpretation [7] of Planck's law of radiation.[2]

To understand the rational of the proposed model it is helpful *not* to expect a concept of space and time on smallest scales, particularly on the question whether they can be assumed continuous or discrete, although the model leans to a discrete structure.

The assessment based on the proposed model: The underlying relation for time relying on *motion* and *state implementation* is (likely) discrete, suggesting *age* as more fundamental than *time*. Regarding Hermann Weyl's Tile Argument [6], I think that Pythagorean Law prevails, not through geometry or a metric, but as a *preserving law* between the above categories. Geometry, needed to define time on our scales, might only evolve on larger scales and dependent on the (dynamic) content of space.

The proposed model is *Informatics* based, driven by interest and not by competence in physics. Nevertheless, I will use terms from physics based on the approach of *Denotational Semantics* [1], keeping the gap between *Syntax* and *Semantics* as close as possible by relating abstract model properties to supposed properties of the real world.

Keywords: Discrete Space, Discrete Time, Twin Paradox, Hermann Weyl's Tile Argument, Einstein's Theories, Sub-Quantum Assumption, Einstein Coefficients

---

*E-mail: pm@KlickModel.org

†No scientific affiliation - author graduated in Informatics at the University of Dortmund

[1] While physics seems to prefer definitions of physical objects in an axiomatic, bottom-up way, Informatics can work from the relation to the object, which is the approach followed here. The definitions of the two more basic categories are considered in the domain of physics. Informatics can contribute the focus on relations between objects, even if their definitions are opaque. Examples are implementations of databases based on Relational Algebra.

[2] maybe of interest, because it adds a 4th coefficient A_{12} with a *defined* density of N_0 of particles in the lower-energy state, which is per construction of the model $N_0 = 0$, which effectively suggests no change to Einstein's formula

Contents

1 Introduction and Prelimineries

1.1 Introduction

This paper and the model are based on the assumption that our world and its physics are described by Einsteins Theories on large scales and by Quantum Theories on small scales. To model possible underlying categories of *time*, I am assuming smaller scales beyond Quantum Theory, in the hope that the designed properties of the model do not contradict the established theories.

For the development of the Klick Model we want to assume the world on three scales, using the term *Macro World* for the world at the scales of Einstein's Relativity Theories. The term *Meso World* refers to our *human-dependant scales* which overlap with the Macro World and reach down to the limits of Quantum Theory. A *Micro World* is assumed which includes the scales of Quantum Theory and reaches down to smallest *hypothetical Sub-Quantum* scales.

We will define two categories, *Motion* and *State Implementation* of abstract particles connected by a preserving relation, the *Klick Relation* (core to the model), based on Pythagorean Triples.

In a first step, we define the *Basic Klick Model* for free abstract particles based on the Klick Relation with just enough properties to calculate the known ageing in the Twin-Paradox. As a next step, the *Extended Klick Model* is added, again on minimalistic assumptions based on encounters of two abstract particles. In a further step, mathematical symmetries to the Basic Model are considered in the *Mirrored Klick Model*, which are very speculative, but falsifiable. We start with some notations for Pythagorean Triples used throughout this paper.

1.2 Notations for Pythagorean Triples

We will write Pythagorean Triples or $\mathcal{T}$ for short as (c, a, b) of three positive integers, assuming $a < b$ unless otherwise noted. We use the letter $\mathcal{P}$ to denote Primitive Pythagorean Triples.

$$\mathcal{T} = \{\ (c, a, b)\ \mid c, a, b \in \mathbb{N}\ \wedge c^2 = a^2 + b^2\} \tag{1.1}$$

Sometimes we want to restrict the set of General or Primitive Triples to a finite set by imposing an upper limit M to the value of c.

$$\mathcal{T}_M = \{\ (c, a, b) \in \mathcal{T}\ \mid c < M\},\ \mathcal{P}_M = \{\ (c, a, b) \in \mathcal{P}\ \mid c < M\} \tag{1.2}$$

Now, we restrict the above sets further to intervals of rational numbers.

$$\mathcal{T}_M[lb, ub]\ = \{\ (c, a, b) \in \mathcal{T}\ \mid\ c < M \wedge\ lb \leqslant b/a \leqslant ub\ \} \tag{1.3}$$

$$\mathcal{T}_M]lb, ub[\ = \{\ (c, a, b) \in \mathcal{T}\ \mid\ c < M \wedge\ lb < b/a < ub\ \} \tag{1.4}$$

In case of $M \to \infty$, we simply use the notations $\mathcal{T}[lb, ub]$ and $\mathcal{P}[lb, ub]$. The algorithms used here to calculate triples to some maximum value of $c < M$ are based on ternary trees [2].

$$\mathcal{T}_M = \mathcal{P}_M\ \cup\ \{\ (ci, ai, bi)\ \mid\ (c, a, b) \in \mathcal{P}\ \wedge\ i \in \mathbb{N} > 1 \wedge ci < M\ \} \tag{1.5}$$

The notation $(*, a, b)$ is used assuming '$*$' as a positive integer $c = \sqrt{a^2 + b^2}$. The set of triples $\{(c, *, *)\}$ stands for triples yielding c. The portion of Primitive Pythagorean Triples up to some maximum value of M converges [3] to $\frac{M}{2\pi}$. Addition of two Triples is only defined if the result is $\in \mathcal{T}$. Otherwise it is undefined[3].

$$(c_1, a_1, b_1) + (c_2, a_2, b_2) = (*, a_1 + a_2, b_1 + b_2) \in \mathcal{T} \tag{1.6}$$

Multiples of the simplest triple $(5, 3, 4)$ are represented by the set $\mathcal{T}[\frac{4}{3}, \frac{4}{3}]$. $\mathcal{T}[1, 1] = \emptyset$ holds, because only positive integers are considered.

[3]This condition is linked to the later definition of *measurements*, which are only defined 'outside' state transitions.

2 Basic Klick Model - based on free abstract Particles

The Basic Klick Model is designed to explain the two different ageing processes in the Twin Paradox. It overloads terms like *particles*, *speed*, or *mass* to describe properties of abstract, isolated (free) Micro World particles in motion. The model is designed to be as basic as possible and should avoid unnecessary assumptions, e.g. about the structure or a geometry of space.

Looking at the Twin Paradox we can simplify the set-up without loss of generality, if we imagine each twin being represented by a particular abstract free particle during the stay and the travel. We will define space of the Micro World as discrete made up by sets of points on the smallest scales in a deliberately simplistic way without suggesting a geometry [4].

For particles, we do not speculate about their structure. We assume them to be described by state information implemented in space and moving through space without speculating about a structure of space. We will assume a 'mechanism' which we will call **Klicks**, which works on space in discrete steps in 'some' uniform way for different subsets of points, the cause for motion and state implementation.

2.1 Two Categories of abstract Particles

We define *abstract* (*conceptual*) particles representing *categories* of particles, counterparts of *categories* of particles of the real world. Their definition depends on the definition of a Micro World *maximum speed*[5], which will follow later with Eq. 2.6.

CF-Particles: The shortcut *CF* stands for *Constantly Fast* moving abstract particles[6]

VF-Particles: The shortcut *VF* stands for *Variably Fast* moving abstract particles below the speed of CF-particles. Per definition VF-particles can approach but never reach the speed of CF-particles. An abstract Micro World particle is either a CF- or a VF-particle.

2.2 Properties of abstract VF- and CF-Particles

To find a model to explain the difference in ageing, we assume a hypothetical structure at smallest possible scales[7], while making minimalistic assumptions, just enough to explain the Twin-Paradox and maybe a bit more. Assumptions on *geometry*, *topology*, *distance*, or *neighbourhood* of real or abstract particles are out of scope. Some analogies from Informatics will be used to explain the design of the model, which should not suggest, that our world is 'computer-model-like'.

Property #1: We only consider VF- and CF-particles.

Let us assume an *Abstract Space* as a set of distinct, discrete *points*. Points *in* space should be understood as points *of* space without a geometry. Compared to computer models, it may help to imagine *space* as an infinite storage for relocatable state information of abstract particles leading to the following assumption.

Property #2: Abstract Space consists of an unlimited amount of discrete points.

A *point* can be compared to a storage location of a computer memory, which can hold a subset of information about some property of a conceptual particle. The next property associates conceptual CF- and VF-particles with subsets of points of an *abstract* space.

[4]The definition of space simply based on sets of discrete points is deliberately incomplete as it avoids the complexities coming with a metric or a geometry. A much bigger problem appears to be to model the dependencies between a conceptual space and the abstract particles living there and the dependencies on their motions. Einstein proved on large scales that geometry depends on the (dynamic) content of space. Why should this be different on smallest scales with smaller distances? This model assumes that Einstein's geometry of space might evolve on larger scales, but it cannot propose an idea, how this might work.

[5]The approach to let terms like *speed* take precedence without previous definitions of *space* and *time* and especially their relation may look unconventional but is core to the 'top-down' approach of the model.

[6]counterparts of particles moving with maximum Meso World speed like photons and neutrinos

[7]compared to our world at much smaller scales than those of Quantum Theory

Property #3: A subset of points can hold the *State Information* of a conceptual particle. Such a subset is referred to as its *Location*. The state information[8] $s \in (D_1 \times D_2, ... \times D_n)$ of an abstract particle can be thought of as a vector of values from some value domains D_i like booleans, natural numbers and others, assuming no limits on precision like in real computers.

Because of the assumed composite structure abstract particles are *not* considered as *point-like*. So far, we introduced abstract CF- and VF particles with a certain state at a certain location. Now, we are looking at the *stepwise* changes of locations and states.[9]

Property #4: A particle's location and/or state information can only change in discrete steps. The stepwise change of a location without changing the state is called **Motion**. The stepwise 'maintenance' of state information without changing the location is called **State Implementation**.

It is essential to the model that a 'frozen' state of a particle cannot exist, i.e. an abstract particle is an endless implementation/development of its state information in discrete steps.

Property #5: Motion/State Implementation of particles through space happens stepwise and simultaneously for all points occupied by a particle. At least the location, or the properties, or both need to change with each step. An abstract particle can never be 'at rest'.

The search for a biological explanation of the different ages of the twins of the Twin Paradox led to the idea to correlate motion and state information of particles. Physical/biological processes could run slower when particles are on the move. This consideration led to the assumption that the maintenance of state information is not 'for free', and that it is connected to motion.

The model does not define *how* a conceptual particle is moving or *how* state information is maintained or encounters with other particles work. Now we want to *count* discrete changes of location and state and correlate their counts.

Property #6: A change of location or state information or both happens for each abstract particle in a same way. We will call these changes **Klicks**. Klicks are expressed as natural numbers and fall into one of two classes: Motion or State Implementation.

The following property is particularly motivated by the Twin Paradox by relating relocation and state information based on Pythagoras' Law as a *preserving relation* and *not* as another application of Euclidean Geometry. We will now relate Klicks required to *relocate* a particle through points in space to Klicks *implementing* the state of the particle.

Property #7: Motion and state propagation of a CF- or VF-particle are *connected* through the **Klick Relation** $K_t^2 = K_r^2 + K_i^2$ which defines a preserving relation between relocating (K_r) and implementing (K_i) Klicks, where $(*, K_i, K_r) \in \mathfrak{T}]1, \frac{4}{3}]$ and $K_r > K_i$ avoiding ambiguous solutions. For simplicity we will sometimes identify a particle x with its Klick Relation.

It is important to understand that the Klick Relation needs not to hold in a *continuous* way, because we do not know, how motion and state implementation work in detail.

Property #8: A **Measurement** is an abstract operation to determine the relocating and implementing klicks K_r and K_i of a particle. It is only defined, if the result is $\in \mathfrak{T}]1, \frac{4}{3}]$.

The model does not define, *how* a particle changes from one valid measurement to the next[10]. The following definition helps modelling a free particle moving undisturbed through space.

Property #9: Conceptual VF- or CF-Particles in steady motion are defined by $\frac{K_r}{K_i} = const.$

Now, we come to the distinction in motion and state implementation patterns between the two categories of abstract particles.

[8]Refering to our world its definition is assumed to be in the domain of physics, on 'larger' scales compatible with Relativity Theory.

[9]State Information on smallest scales is considered as non-static, resembling refresh cycles for computer memory to maintain storage information or humans refreshing their memory during deep sleep, i.e. never being 'at rest'.

[10]This resembles the view on CPU instructions executed by micro code. Measurements are only meaningful between *completed* CPU instructions.

Property #10: The Klick Relation for VF-Particles is defined through $(*, K_i, K_r) \in \mathcal{T}]1, \frac{4}{3}[$. Each VF-Particle is of a *certain kind*, being a multiple of a 'base relation' $\in \mathcal{P}]1, \frac{4}{3}[$.

Property #11: The Klick Relation for CF-Particles is defined through $(*, K_i, K_r) \in \mathcal{T}[\frac{4}{3}, \frac{4}{3}]$.

2.3 The Klick Relation for VF- and CF-Particles

The following diagram sketches the Klick Relations for VF- and CF-Particles.

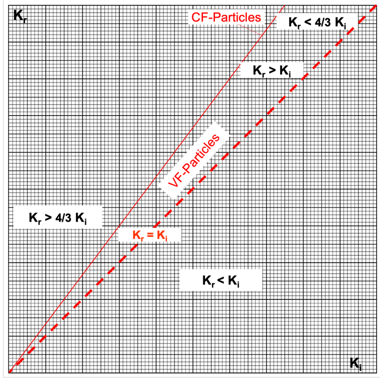


Figure 1: Klick Relations of VF- and CF-Particles in $\mathcal{T}]1, \frac{4}{3}]$.

'Speed' of relocation is traded for 'speed' of maintenance of state information for VF- and CF-Particles. The more Klicks are needed to move a particle, the less Klicks are available for the implementation of its state information. No metric was needed to come to this result. A quantitative test will follow in chapter 5.1.

2.4 Overloading Model Properties with Terms from our World

We will now extend our abstract model of a hypothetical Micro World with further attributes, reusing / overloading terms from our Meso World. The following terms are only applicable to isolated (free), simple, non-composite VF-Particles.

Creation and Absorption: The term *creation* for abstract particles is borrowed from the Meso World and refers to the *creation of a particle* in a process. We use the term *absorption* as the counterpart of creation, i.e., when an abstract particle 'exits' a state of steady motion.

Measurement: The term *measurement* is meant in an abstract operation to determine the relocating and/or implementing klicks of an abstract particle.

Age: The age of an abstract particle in steady motion is defined as the number of implementing Klicks, i.e., K_i, from its creation to a measurement. Each particle has its own age based on *counting* the implementing klicks since its creation.

$$Age = \sum_{creation}^{measurement} K_i \tag{2.1}$$

Duration between Measurements of a Particle: A duration is the difference between two measurements of ages of the same particle in steady motion.

$$Duration = \sum_{creation}^{measurement_2} K_i - \sum_{creation}^{measurement_1} K_i \tag{2.2}$$

Horizon: The horizon is the maximum number of relocations an abstract particle can have since its creation.

$$Horizon = \sum_{creation}^{measurement} K_r \tag{2.3}$$

Speed: In our Meso World speed is defined as a derived quantity based on geometry and time. For our abstract model we take a different approach by considering abstract particles dominated by relocating over implementing klicks intuitively as *faster*. The speed of an abstract particle in steady motion is defined on the basis of its Klick Relation.

$$\mathit{Speed} = \frac{K_r}{K_i} \ . \tag{2.4}$$

Faster and Slower: If we relate implementing klicks to the amount of information which is transported by relocating klicks, it will fit our intuition that the transport of more information with the same transport capacity results in a lower speed. For two separate free particles x and y in steady motion, we define x as **faster** compared to y if the following condition holds:[11]

$$x \ \mathit{faster\ than}\ y \quad \equiv \quad \frac{K_r(x)}{K_i(x)} > \frac{K_r(y)}{K_i(y)} \ . \tag{2.5}$$

Speed-Limit: Based on the previous definitions for CF-Particles we assume a speed limit defined for abstract particles with klick relations of the form $(5K_t, 3K_r, 4K_i) \in PT \ for \ n > 0$.

$$\mathit{Speed\text{-}Limit} = \frac{K_r}{K_i} = \frac{4}{3} \tag{2.6}$$

The next definition is very speculative, simply following from mathematical symmetry. The term *mass* should be understood as *mass-related*.

Mass-/Motion Ratio and its Limit: The mass of a conceptual Particle might be related to the amount of (state) information a Micro World particle is carrying. More mass means more information. The *mass-motion-ratio* of a free abstract particle in steady motion is defined based on symmetry to the definition of *speed*, including a definition of *faster*.

$$\mathit{Mass\text{-}Motion\text{-}Ratio} = \frac{K_i}{K_r} \ . \tag{2.7}$$

$$\mathit{Mass\text{-}Motion\text{-}Ratio\text{-}Limit} = \frac{3}{4} \ . \tag{2.8}$$

$$x \ \mathit{is\ more\ mass\text{-}related\ than}\ y \quad \equiv \quad \frac{K_i(x)}{K_r(x)} > \frac{K_i(y)}{K_r(y)} \ . \tag{2.9}$$

[11] defining a partial order on the speeds of conceptual particles

3 Extended Klick Model - based on Two-Particle Events

So far, we considered free abstract VF- and CF-particles $\in \mathfrak{T}]1, \frac{4}{3}]$ in steady motion. Now, we will look at *encounters* between *two* abstract particles, leading to the *key definition* of Eq.3.1 below.

3.1 CF-Packets - The smallest CF-Particle and its Multiples

The following properties follow from the concept of Pythagorean Triples.

Property #12: There is a smallest Klick Relation of a CF-Particle, which is $\hat{h} = (5, 3, 4)$.

Property #13: *CF-Packets* are abstract particles with a klick relation of $n\hat{h} = (5n, 3n, 4n) \in \mathfrak{T}[\frac{4}{3}, \frac{4}{3}]$ with $n \in \mathbb{N}, n > 0$.

3.2 Coupling VF-Particles and CF-Packets

The extended model will assume that CF-Particles can only connect to a VF-Particle, if the aggregate conserves the Klick Relation. The 'coupling process' is considered as opaque. We are only interested in the Klick Relations binding motion and state implementation as Pythagorean Triples for the particles *before* coupling and *after* resulting into some new aggregate.

3.2.1 States resulting from Coupling Events

Now we combine a VF-Particle with one or more CF-Packets while preserving the Klick Relation of the result.

Property #14: A VF-Particle x, *coupling* with a CF-Packet is called in some *state*, if there is a *CF-Packet* with some factor $f \geqslant 0$ and limit M such that the following condition holds.

$$x \in \mathfrak{T}_M]1, \frac{4}{3}[\ \wedge \ f\hat{h} \in \mathfrak{T}_M[\frac{4}{3}, \frac{4}{3}] \implies x + f\hat{h} \in \mathfrak{T}_M]1, \frac{4}{3}[. \tag{3.1}$$

For $f > 0$, the values of f, let us call them f_i, can be ordered into packets according to the 'C-value' of $x + f\hat{h}$. Each factor f_i can then be expressed as a sum of n CF-Packets, i.e. $f_i = \sum_{i=1}^{i \leqslant n} f_{pkg_i}$.

The number of CF-Packets, which can couple with a certain VF-Particle depends on f, the maximum number of which in turn depends on M. For a given value of M and a given VF-Particle there can exist zero or more CF-Packets.

For example, if $M = 800000$ and $x = (797, 555, 572)$, there are 2 factors fulfilling the above condition: $f_1 = 265$ and $f_2 = 265 + 2117 = 2382$ with CF-Packets $f_{pkg_1} = 265$ and $f_{pkg_2} = 2117$. The sequence of factors f and also f_{pkg} is not continuous. Instead it seems that the deltas between packages f_{pkg_i} steeply increase with M when looking at bigger value ranges.[12]

3.2.2 Isolated States, Ground States and Excited States of Abstract Particles

A VF-Particle is called in an *Isolated State* with regard to Klick Relations $\mathfrak{T}_M]1, \frac{4}{3}[$ limited by a maximum value of M according to Eq. 3.1, if the equation is *only* fulfilled for $f = 0$. Otherwise there exist CF-Packets with f_{pkg_i} with $i > 0$. VF-Particles are called in a *Ground State*, if the equation is fulfilled for f_{pkg_i} with $i = 0$ while values $i > 0$ exist. Otherwise VF-Particles are called in an *Excited State*.

This definition leaves open, whether there could be particles in a ground state with no excited states for any value of M. It seems that there is an infinite set of excited states for each ground state with sharply decreasing occurrences, i.e. with no upper limit to f in the above definition. This assessment is only based on some computational tests. Note, that Isolated States and Ground States might be *promoted* to Excited States with increasing M.

[12] see algorithm to generate this and further VF-/CF-correlations in AppendixB

3.2.3 Generic Ensembles of States

Now, we will construct a series of finite sets, which we want to call *Generic Ensembles* dependant on some natural number M as some upper bound for an enumeration of Pythagorean Triples. Each set is based on VF-Particles $\in \mathcal{P}_M]1, \frac{4}{3}[$ in ground states, for which we calculate their excited states $\subset \mathcal{T}_M]1, \frac{4}{3}[$ by applying CF-Packets to them. The choice to look only at Primitive Pythagorean Triples as ground states is fairly arbitrarily, but convenient for the simplicity of its construction.

For the tables listed further down, which are based on the following definition, it is important to understand, that we want to associate each ground state with *all* excited states, i.e. one VF-Particle in its ground state is combined with multiple excited states (although limited by M).

Property #15: A Generic Set $\mathcal{E}$ of excited states limited by M results from VF-Particles encountered by CF-Packets yielding VF-Particles $\subset \mathcal{T}_M]1, \frac{4}{3}[$, if the following condition holds for at least one factor $n \in \mathbb{N}, n > 0$.

$$\mathcal{E}_M = \{ \ (c, a, b) \in \mathcal{T}_M]1, \frac{4}{3}[\ \mid \exists (c', a', b') \in \mathcal{P}_M]1, \frac{4}{3}[\wedge a = a' + 3n \ \wedge \ b = b' + 4n \ \} \tag{3.2}$$

The generic set $\mathcal{E}_M$ of excited states is based on a generic set of VF-Particles in their ground state for $n > 0$ as follows.

$$\mathcal{G}_M = \{ \ (c, a, b) \in \mathcal{P}_M]1, \frac{4}{3}[\ \mid \exists (c', a', b') \in \mathcal{E}_M \ \wedge \ a' = a + 3n \ \wedge \ b' = b + 4n \ \} \tag{3.3}$$

The generic set $\mathcal{I}_M$ of isolated states is based on a generic set of VF-Particles in their ground state for $n = 0$ as follows.

$$\mathcal{I}_M = \{ \ (c, a, b) \in \mathcal{P}_M]1, \frac{4}{3}[\ \mid \nexists (c', a', b') \in \mathcal{E}_M \ \wedge \ a' = a + 3n \ \wedge \ b' = b + 4n \ \} \tag{3.4}$$

One could decide here to define a ground state as the union $\mathcal{G}_M \cup \mathcal{I}_M$, because elements of $\mathcal{I}_M$ could couple with CF-Particles with an increased value of M. But, we want to decide here to segregate the 'logical ground states' into the sets $\mathcal{G}_M$ and $\mathcal{I}_M$.

Obviously $\mathcal{E}_M \subseteq \mathcal{E}_N$ for integers $M < N$ and $\mathcal{E}_M \cap \mathcal{G}_M \cap \mathcal{I}_M = \emptyset$. Here is an example for $\mathcal{E}_{800000}$.

For $(35113, 24745, 24912) \in \mathcal{G}_{800000}$ there are CF-Packets, i.e. multiples of $\hat{h}$, which are $n_1 = 42, n_2 = 3648, n_3 = 5622, n_4 = 32304, n_5 = 2094, n_6 = 26670$, such that $(*, 24745 + 3 * \sum_{j=1}^{j \leqslant i} n_i, \ 24912 + 4 * \sum_{j=1}^{j \leqslant i} n_i) \in \mathcal{E}_M$.

Only with $\mathcal{E}_{3200000}$ we can get $n_7 = 457002$ with $(*, 24745 + 3 * \sum_{j=1}^{j \leqslant 7} n_7, \ 24912 + 4 * \sum_{j=1}^{j \leqslant 7} n_7) \in \mathcal{E}_M$. Looking at increasing ensembles up to $\mathcal{E}_{25600000}$ a value following n_7 cannot be provided here, maybe because of exponential growth and/or lack of computing power of a conventional laptop.

Let us look now at examples of Generic Ensembles with increasing values of M, which include the above example.

3.2.4 Modelling State Transitions in Generic Ensembles of States

So far, we described a static picture of Generic Ensembles. The following extension assumes a 'dynamic' through attachment and detachment of CF-Particles in these sets, while preserving the Klick Relation. Let us assume two types of attachments/detachments to be dependent on the number of single or multiple occurrences of the same CF-Particle in the set - expressed through the values $nA21$ and $nB21$. The columns of the following tables are described here.

N_0	VF-Particles in an isolated state, not able to transition to a lower/higher state
N_1	VF-Particles with potential to transition to a higher excited state
N_2	VF-Particles with potential to transition to a lower excited state or the ground state
$nA21$	Unique CF-Particles (coupling only with one VF-Particle as in Eq. 3.1)
$nB21$	Non-unique CF-Particles (coupling with > 1 VF-Particles as in Eq. 3.1)
$nA12$	CF-Particles coupling with VF-Particles in an isolated state, i.e. $nA12 = 0$
$nB12$	All occurrences of f in Eq. 3.1
r_n	Calculated value from the above values with the relation shown below.

Based on the definitions Eq. 3.2, 3.3 and 3.4 the values of N_0, N_1 and $N2$ can be expressed as follows.

$$N_0 = |\mathcal{I}_M|, \quad N_1 = |\mathcal{G}_M|, \quad N_2 = |\mathcal{E}_M| \tag{3.5}$$

With

$$r_n = \frac{nA_{21} * N_2}{nB_{12} * N_1 - nB_{21} * N_2} \tag{3.6}$$

we correlate the above values as follows.

$$nA_{12} * N_0 \ + \ nB_{12} * N_1 * r_n = nA_{21} * N_2 \ + \ nB_{21} * N_2 * r_n \tag{3.7}$$

Since $nA_{12} = 0$, we get

$$nB_{12} * N_1 * r_n = N_2 * (nA_{21} \ + \ nB_{21} * r_n) \tag{3.8}$$

3.2.5 Examples of Generic Ensembles

The following tables are the result of enumerating ensembles $\mathcal{E}_M$, $\mathcal{G}_M$ and $\mathcal{I}_M$ for increasing values of M, including some more parameters[13]. Table 1 has no limit on the number of levels.[14]

Table 1: Generic Ensembles with n Levels

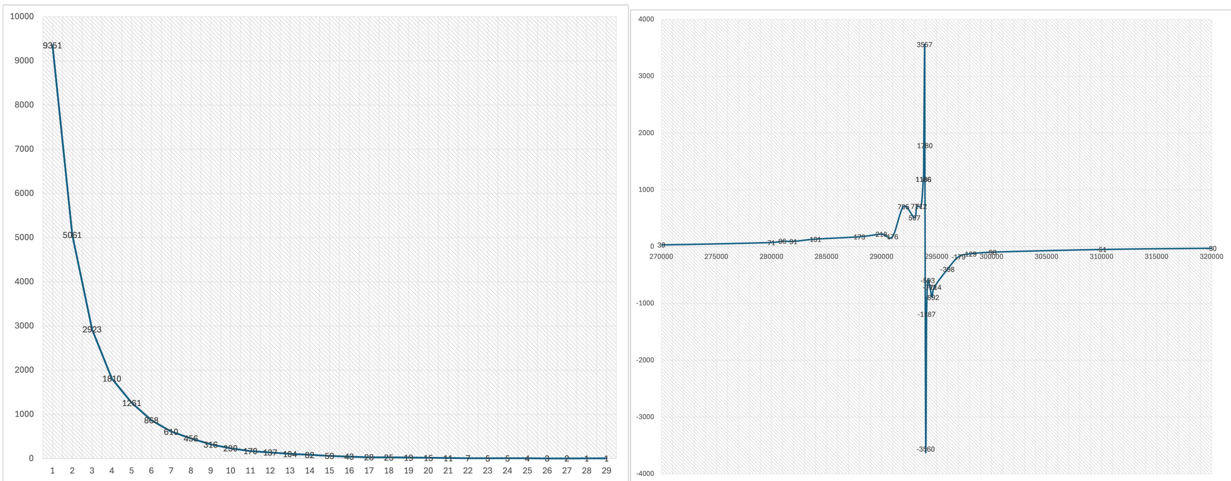
M	N_0	N_1	N_2	$nA21$	$nB21$	$nB12$	r_n	$nUniqPkg$	$pkgAvg$	$nLevel$	g
$1*10^5$	2021	856	1544	1053	491	1544	2.88493	1284	3805	10	4
$2*10^5$	3786	1960	4006	2375	1631	4006	7.21884	3106	6808	14	5
$4*10^5$	7194	4305	9792	4913	4879	9792	-8.55923	6999	12559	23	6
$8*10^5$	13652	9351	23607	9846	13761	23607	-2.23265	15331	22429	29	8
$16*10^5$	26205	19805	55964	19328	36636	55964	-1.14836	32893	40850	38	11
$32*10^5$	50446	41563	131327	36977	94350	131327	-0.70049	68891	74095	59	15
$64*10^5$	97439	86578	304612	69988	234624	304612	-0.47274	141644	135452	79	22
$128*10^5$	189160	178893	698915	132603	566312	698915	-0.34227	286987	247370	105	25

Table 2 only considers *two* levels, which are the ground states and the first excited states. So, the values are limited by M *and* $nLevel = 1$, i.e. by $f_{pkg_i} \leqslant f_1$ in Eq. 3.1. The number of excited

Table 2: Generic Ensembles with just 2 Levels - Ground and first Excited State

M	N_0	N_1	N_2	$nA21$	$nB21$	$nB12$	r_n	$nUniqPkg$	$pkgAvg$	$nLevel$	g
$1*10^5$	2021	856	856	731	125	856	1.00000	792	3919	1	3
$2*10^5$	3786	1960	1960	1579	381	1960	1.00000	1759	7239	1	4
$4*10^5$	7194	4305	4305	3334	971	4305	1.00000	3788	14154	1	5
$8*10^5$	13652	9351	9351	6922	2429	9351	1.00000	8040	26135	1	5
$16*10^5$	26205	19805	19805	14066	5739	19805	1.00000	16633	49435	1	6
$32*10^5$	50446	41563	41563	28116	13447	41563	1.00000	34025	94688	1	8
$64*10^5$	97439	86578	86578	56162	30416	86578	1.00000	69233	182598	1	9
$128*10^5$	189160	178893	178893	110928	67965	178893	1.00000	139450	350464	1	11

states per level decreases exponentially in Figure 2a with 9351 for the first level followed by 5061 for the next level and so on until 1 for the highest level of 29.



(a) Excited States per Level @ $M = 8*10^5$ (b) Development of r_n with a spike at about $\mathcal{E}_{294016}$

Figure 2: Excited State and r_n Distributions for Ensembles

Figure 2b shows the development of r_n, a rational number, which seems to be independent of enumerations of ensembles. It displays an asymptotic increase to a positive constant, then a spike to a negative constant followed by an asymptotic increase to zero at about $M \approx 294016$.

[13] $nUniqPkg$: Number of unique CF-Packets; $pkgAvg$: Average value of a CF-Packet per excited state; $nLevel$: Maximum number of excited states per ground state in $\mathcal{E}_M$; g: Maximum number of states sharing the same value of f in Eq. 3.1 with regard to $\mathcal{E}_M$.

[14] The value of $nLevel$ depends on the underlying set of Pythagorean Triples with limit M. The values in Table 1 are limited through $\mathcal{T}_M]1, \frac{4}{3}[$ and range between 1..105.

3.2.6 Probability of VF-Particles coupling with CF-Packets

The application of the Basic Klick Model to the Twin-Paradox or the consideration of structural similarities to interactions between matter and radiation in our world do not have the weight of a proof. This would be different, if one could derive a physical law or constant from the model, e.g. using Generic Ensembles. The following is not a solution, only an idea for an approach.

Let us explore the idea to relate the Fine-Structure Constant α to Generic Ensembles of the type $\mathcal{P}_M]1, \frac{4}{3}[$ and the quantitive relations between CF- to VF-Particles, which may contribute to the probability of their interactions. The number of CF-Particles is $M/5$. While the number N of VF-Particles for $\mathcal{P}]0, \infty[$ is determined by the formula $N = M/(2\pi)$, there is only a heuristically determined[15] number N of VF-Particles for $\mathcal{P}_M]1, \frac{4}{3}[$ available here as a fixed factor of $N = |\mathcal{P}_M]1, \frac{4}{3}[| \simeq M/34.7774$ based on Table 5. N corresponds to $N = N_0 + N_1$ in Table 1.

$$\frac{M/5}{N} * f = \alpha^{-1} \tag{3.9}$$

A value of $f \simeq 19.70186232$ would fulfil this equation, but unfortunately would only be a constructed value by simply correlating results of Table 5 with the desired result of α. If f could be derived[16] from mathematical formulas or from statistics, the correlation in Eq. 3.9 would become interesting.

Seen from the perspective of the unsolved problem of finding a (purely mathematical) formula for α, the discussed approach looks a bit like exchanging an unknown problem by another unknown problem, now trying to find a formula for ≈ 19.70186232. [17] This seems not to be a solution.

Here is a further idea. PPTs are more convenient to find mathematical relations compared to PTs. But the model is assuming PTs to map to relations of the real world, so we are more interested in PTs. The term $\frac{M/5}{N}$ on the left-hand side of Eq. 3.9 relates to $\mathcal{P}]1, \frac{4}{3}[$, while the right-hand side is relating to $\mathcal{T}]1, \frac{4}{3}[$. The value of f would need to combine these two sets, but how?

For instance f could be the product of a constant, like $2\pi^2$, modified by a *growth factor* f', which correlates $\mathcal{P}]1, \frac{4}{3}[$ and $\mathcal{T}]1, \frac{4}{3}[$. Lacking a better idea f', if taken as ratio between $\pi_{\mathcal{P}]1,\frac{4}{3}[}$ to $\pi_{\mathcal{T}]1,\frac{4}{3}[}$ based on Table 5 and Table 6 as $f' = 1.00178898$ will lead to the modified equation with a bias to α of 1.00006849 - still not good enough to be considered as a candidate for a solution.

$$\frac{M/5}{N} * 2\pi^2 * f' = \alpha^{-1} \tag{3.10}$$

Eq. 3.10 should instead be understood as an idea to encourage better proposals. The challenge is to relate the development of $\mathcal{T}_M]1, \frac{4}{3}[$ to our world with increasing M. Maybe the coupling condition based on '+' in Eq. 3.1 is also too simplistic. But no better ideas[18] can be put forward here.

Still, I would also like to include a 'reverse-engineered' idea based on attempts to find a mathematical formula for α. If one of these proposals[19] could be based in a meaningful way on Pythagorean Triples as proposed by Eq. 3.1, this could possibly add credibility to the proposed model.

[15] see A101929 - Number of Pythagorean triples with hypotenuse $< 10^n$ [9] - based on an algorithm

[16] like $2 * \pi^2$, which is close, although not close enough

[17] I think, if someone will ever find a formula for α, it will be related to Pythagorean Triples in some way, not related to *space* and *time*, but to some underlying categories, whatever they are.

[18] An example of another preserving operations, $(x_1, y_1, z_1) \circ (x_2, y_2, z_2) = (x_1x_2, y_1z_2 + y_2z_1, y_1y_2 + z_1z_2)$, which is not further considered here, is discussed in [10].

[19] There are several proposals to derive α from some mathematical formula. One of them is the so-called 'deVries formula' [11]. Of course, precision alone is not a prove. A broken railway clock gives an almost perfect time of almost infinite precision, two times per day. But a formula related to a cause in a meaningful way could make a difference.

3.3 Events between VF-Particles

Events between VF-Particles and CF-Particles discussed in the previous chapters are more important to test the plausibility of the Klick Model than events between VF-Particles, which are discussed here. Their consideration should at least not lead to contradictions.

For encounters of VF-Particles we will limit our considerations to *binary events*, where pairs of VF-particles either merge into some 'new' target particle, or where a particle is split into two VF-particles. We are interested in motion/state propagation patterns of VF-particles x and y either joining or splitting, while preserving the total number of Klicks as subsets of $\mathfrak{T}]1, \frac{4}{3}[$ under a "+" operation [20]. This operation is defined if the following preserving conditions hold:

$$K_t^2(x) + K_t^2(y) = K_t^2(z) \;\mid\; x, y, z \in \mathfrak{T}]1, \frac{4}{3}[. \tag{3.11}$$

$$x, y \in \mathfrak{T}]1, \frac{4}{3}[\implies (*, K_r(x) + K_r(y), K_i(x) + K_i(y)) = z \in \mathfrak{T}]1, \frac{4}{3}[. \tag{3.12}$$

Note that this definition allows for an abundance of solutions, which are inherently ambiguous. For instance, $(45, 27, 36)$ can be the result of $(25, 15, 20) + (20, 12, 16)$, or $(35, 21, 28) + (10, 6, 8)$, or $(30, 18, 24) + (15, 9, 12)$, or $(40, 24, 32) + (5, 3, 4)$. Like $(10, 6, 8) + (5, 3, 4) \notin \mathfrak{T}]1, \frac{4}{3}[$, not all additions land in $\mathfrak{T}]1, \frac{4}{3}[$, which leads us to the definition of *compatibility* of conceptual particles.

We will now define a 'Compatibility Condition' for Two-Particle Events. We assume that two abstract particles can only interact if they follow the conservation principle of 3.12. We call such particles *compatible* with regard to their klick relations.

Property #16: Two abstract particles x and y can only join into a new object z under the operation '+' if they are *klick compatible*. They are compatible with regard to their motion/state propagation if the following condition holds for the target particle.

$$x, y \in \mathfrak{T}]1, \frac{4}{3}[\implies x + y = z \in \mathfrak{T}]1, \frac{4}{3}[. \tag{3.13}$$

No suggestions can be made here to measure *compatibility*. Maybe this is possible based on the previously explained Generic Ensembles.

[20] which corresponds to the addition of Gaussian Integers or Complex Numbers

4 Mirrored Klick Model - based on mathematical Symmetries

The Mirrored Klick Model is the result of mirroring the Klick Relation $\mathfrak{T}]1, \frac{4}{3}]$ for VF- and CF-Particles to $\mathfrak{T}[\frac{3}{4}, 1[$ based on symmetry relative to an axis defined by $K_r = K_i$. While relocating klicks dominate implementing klicks in the Basic Klick Model, it is the other way round in the mirrored model. If not otherwise noted the Klick Model's assumptions regarding space, state propagation, speed, age, horizon, and so on will re-apply to the Mirrored Klick Model trading relocating for implementing klicks.

4.1 Categories of Particles mirroring CF- and VF-Particles

For the Mirrored Model, we will distinguish between minimum mass instead of maximum speed, considering only free particles in steady motion. The term *mass* in relation to *motion* has been introduced with Eq. 2.7 and is a purely abstract attribute of the model. We will postulate now counterparts to the conceptual classes of CF- and VF-Particles.

CM-Particles: The shortcut *CM* stands for *Constant in Mass* for particles dominated by K_i. We will assume a relation defining a conceptual *Micro World Minimum Mass* for CM-particles. Collections of CM-particles are described by the set $\mathfrak{T}[\frac{3}{4}, \frac{3}{4}]$.

$$\textit{Klick Relations of CM-particles} \in \mathfrak{T}[\frac{3}{4}, \frac{3}{4}] \tag{4.1}$$

$$\textit{Motion-Mass-Ratio-Limit} = \frac{K_r}{K_i} = \frac{3}{4} \tag{4.2}$$

VM-Particles: The shortcut *VM* stands for particles *Varying in Mass* with $\frac{K_r}{K_i}$ differing from $\frac{3}{4}$. The motion/state propagation patterns of these hypothetical particles are described by the set

$$\textit{Klick relations of VM-particles} \in \mathfrak{T}]\frac{3}{4}, 1[\tag{4.3}$$

4.2 A combined View of the Basic Klick Model and its Mirror

Figure 3 sketches the combination of both models with abstract particles conforming to the set $\mathfrak{T}[\frac{3}{4}, \frac{4}{3}]$. The symmetry line defined by the relation $K_r = K_i$ corresponds to the empty set.

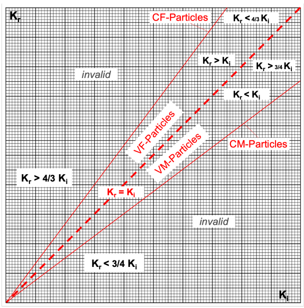


Figure 3: Combined View of the Klick Model and its Mirror.

In the hypothetical 'Speed/Mass Mirrored Model' the roles of K_r and K_i are swapped. CM- and VM-particles take over the role of CF- and VF-particles, suggesting a massive counterpart of the CF-Particle, which I would call *Bagongs*[21], should they exist at all.

The Klick Model itself is very speculative. The proposed Mirrored Model goes on-top of these speculations. As weird as it may sound, the Mirrored Model makes the most precise prediction of the model, which should be testable in the real world - as outlined in the next chapter.

[21]The name *Bagong* is chosen as counterparts of photons after a clumsy character of Indonesian Wayang Kulit.

5 Looking for Tests of the Klick Model

Whether a model or theory is useful to understand our real world depends on the question, whether it can predict results from real observations, experiments or proven theories.

5.1 Applying the Basic Klick Model to the Twin-Paradox

The Klick Relation combines relocating and implementing klicks in a preserving relation, summarised by $K_t^2 = K_r^2 + K_i^2$. In other words: Speed of relocation is traded for speed of change of state information and vice versa. The more Klicks are needed to move a particle, the less Klicks are available for the implementation of its state information and vice versa.

Without loss of generality, we will assume twins, one staying on our planet, and a travelling one who travels extremely fast (close to 'a' maximum speed) and who joins the other twin after years, neglecting the accelerating and deceleration phases. For the staying and travelling twins s and t we assume the following start conditions.[22]

$$K_t(s)^2 = K_r(s)^2 + K_i(s)^2 \tag{5.1}$$

$$K_t(t)^2 = K_r(t)^2 + K_i(t)^2 \tag{5.2}$$

$$K_r(s) = K_i(s) = K_r(t) = K_i(t) = 0 \ . \tag{5.3}$$

The following condition holds from the start to the end with the reunion of the twins:

$$K_t(t) = K_t(s) \ \textit{with} \ \sum_{start}^{end}(K_r(t) + K_i(t)) = \sum_{start}^{end}(K_r(s) + K_i(s)) \ . \tag{5.4}$$

Because t is travelling *extremely* fast compared t, we make two further observations:

$$K_r(t) \gg K_r(s) \ , \tag{5.5}$$

$$K_r(s) \to 0 \text{ compared to } K_r(t) \ . \tag{5.6}$$

As a next step, we will combine the above equations while focusing on the correlation between K_r and K_i and Eq. 5.4, setting

$$K_t(s) = K_t(t) = 1. \tag{5.7}$$

With $K_r(s) \to 0$ in Eq. 5.1 we yield

$$K_t(s)^2 \to K_i(s)^2 \ , \text{ i.e. } K_i(s) \to 1 \ . \tag{5.8}$$

Combining the two Klick Relations from Eq. 5.1 and 5.2 for the reunion of the twins, we yield

$$K_i(s)^2 \to K_r(t)^2 + K_i(t)^2 \ , \text{ which is the same as } K_i(t)^2 \to K_i(s)^2 - K_r(t)^2 \ . \tag{5.9}$$

With $K_i(s)^2 \to 1$ we yield:

$$K_i(t) \to \sqrt{1 - K_r(t)^2} \ . \tag{5.10}$$

Example: A twin moving at 80% of maximum speed will experience the following proportion of K_i (corresponding to the *biological/atomic age*), no matter how a maximum speed is defined:

$$K_i = \sqrt{1 - 0.8^2} = 0.6 \ . \tag{5.11}$$

When the twins meet after 10 earth years, the travelling twin will only be $10 * 0.6 = 6$ years older, conforming to Einstein's Special Relativity Theory. No metric was needed to come to this result.

[22] simplifying the starting condition by a triple $(0, 0, 0)$, which is, strictly speaking, not a valid Klick Relation

5.2 Applying the Klick Relation to the Energy-Momentum Relation

Looking for arguments for the Klick Model the following speculative consideration relates the Klick Model to the *Energy-Momentum Relation* [4]. The two relations are of the following forms.

$$K_t^2 = K_r^2 + K_i^2 \tag{5.12}$$

$$E^2 = p^2\, c^2 + m_0^2\, c^4 \tag{5.13}$$

By only looking at the forms of the equations we yield the following correlations:

$$K_t^2 \sim E^2 \quad \widehat{=} \quad K_t \sim E \tag{5.14}$$

$$K_r^2 \sim p^2\, c^2 \quad \widehat{=} \quad K_r \sim p\, c \tag{5.15}$$

$$K_i^2 \sim m_0^2\, c^4 \quad \widehat{=} \quad K_i \sim m_0\, c^2 \tag{5.16}$$

Should this correlation exist, it would mean:

- Motion, expressed through K_r, is connected to p, which involves a direction and an impact.
- State information, expressed through K_i, is connected to the rest mass m_0.
- Setting $p = m_0 = 1$ would suggest, that implementing klicks would dominate relocating klicks because of the factor c^2 in Eq. 5.16 compared to c in Eq. 5.15.
- Relating Eq. 2.4 to the Energy-Momentum-Equation suggests that Micro World Speed could be expressed by the relation $\frac{p}{m_0}$ with c set to 1. Relating Eq. 2.7 suggests that mass in the Micro World could be expressed as $\frac{m_0}{p}$ with c set to 1.

Both preserving relations can be correlated - at least based on their form. The following calculation of the Twin-Paradox based on the Energy-Momentum-Relation is meant as a plausibility check for the Klick Model. For the staying twin s and the travelling twin t we define:

$$E_s^2 = p_s^2\, c^2 + m_{0s}^2\, c^4 \tag{5.17}$$

$$E_t^2 = p_t^2\, c^2 + m_{0t}^2\, c^4 \tag{5.18}$$

The following condition is assumed to hold from the start to the end with the reunion of the twins, if factors like gravity, rotation or intensive local motion can be neglected:

$$E_s = E_t \tag{5.19}$$

Because the travelling twin is travelling *extremely* fast, we make two further observations:

$$p_t \gg p_s \quad \text{and} \quad p_s \to 0 \quad \text{compared to} \quad p_t\ . \tag{5.20}$$

As a next step, we combine the above equations while focusing on the correlation[23] between $p\, c$ and $m_0\, c^2$ by setting $E_s = E_t = 1$. With $p_s \to 0$ in Eq. 5.17, we yield

$$E_s^2 \;\to\; m_{0s}^2\, c^4 \quad \text{, i.e. } m_{0s}\, c^2 \to 1\ . \tag{5.21}$$

Combining the Energy-Momentum Relations of the twins from 5.17 and 5.18 we yield

$$m_{0s}^2\, c^4 \;\to\; p_t^2\, c^2 + m_{0t}^2\, c^4 \text{ , which is the same as } m_{0t}^2\, c^4 \;\to\; m_{0s}^2\, c^4 - p_t^2\, c^2\ . \tag{5.22}$$

With $m_{0s}\, c^2 \to 1$ we yield:

$$m_{0t}\, c^2 \quad \to \quad \sqrt{1 - p_t^2\, c^2}\ . \tag{5.23}$$

Should $p_t^2\, c^2$ be related to the speed of a twin travelling at 80% of the maximum speed and $m_{0t}\, c^2$ to its ageing, it would experience[24] the following percentage of ageing.

$$m_{0t}\, c^2 = \sqrt{1 - 0.8^2} = 0.6\ . \tag{5.24}$$

[23] The relation $p_s^2\, c^2 / m_{0s}^2\, c^4$ is dimensionless - further supporting the analogy to the Klick Relation.

[24] Regarding information processing the travelling twin does not have a competitive advantage. If both twins would read the same books during the trip, the travelling twin would only have read 60% of them compared to its sibling.

5.3 Predicting the Non-Existence of *massless* and *motionless* Particles

Should it be possible to relate the Klick Model, which is based on the definition of Pythagorean Triples, to the real world, it would support the following statements.

- Massless particles *cannot* exist.
- Particles without motion *cannot* exist.

At first glance, the predictions look like a chance to falsify the Klick Model. But the problem could be the limit of the achievable precision of physical measurements.[25]

5.4 Explaining the exactly same speed of Photons and Neutrinos

The proposed model suggests that the maximum speed of abstract particles only depends on the *ratio of relocating and implementing klicks.* Mapped to the physical world, this ratio could be the same for photons and neutrinos - not being dependent on *mass*.

5.5 Possible Method to falsify the Mirrored Klick Model

While it is a bold idea to suggest CM-Particles or *Bagongs* as counterparts of *Photons* in Figure 4, it allows a precise prediction of their fixed maximum speed of $\frac{9}{16} = 0.5625$ of c, i.e. $168,633,257.625\ \frac{m}{s}$.

Does data exist[26] for instance for the speed difference of dark energy and dark matter departing from the centre of a cosmic event like from a supernova? If such data exists and this data would hint to some constant factor deviating from 0.5625, it would effectively falsify the idea about the Mirrored Model and possibly about the Klick Model as such.

The following diagram combines our world and a supposed 'dark world' in one picture. The bold dotted line marked by $K_r = K_i$ corresponds to the empty set.

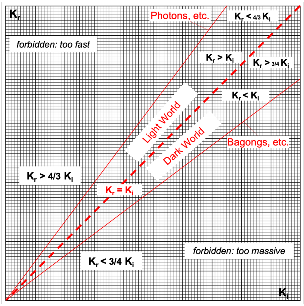


Figure 4: Motion/State Propagation in a hypothetical World limited by Photons and Bagongs.

If related to our world, this picture suggests that particles of the light and the dark world can only interact with particles of the same domain. If this is true, the only common ground between the *Light World* and the *Dark World* is *Space*, i.e., processes can only be detected in either world through effects of changes of space (through gravity).

But how could both worlds share the same space and how could it be tested? Maybe space has a strange 'fabric'. Imagining a chessboard with light and black squares and moving white and black bishops (not a true proposal) may help our imagination.

[25] Regarding the Energy-Momentum-Relation the Klick Model would suggest $E \to mc^2$ instead of $E = mc^2$, which indeed might be a problem to defend the design of the model.

[26] Some data seem to exist, e.g. [5], but I do not have a sufficient understanding to include these works here.

6 Conclusions

Time is whatever physics defines it on our scales. The question, whether *time* is a fundamental category would normally call for a proof by contradiction. To find a constructive approach in form of the proposed model appears more difficult, but could provide more insights, like *age* might be more fundamental than *time* on smallest scales.

Although I think that space is discrete, which is not essential to the model, it paid off to avoid considerations about the structure of space. Looking at Hermann Weyl's Tile Argument I think that Pythagoras' Law is correct, not because of an implied geometry or metric but as a preserving law (between motion and states of particles).

Einstein's Theories proved that the motions of particles in space are *content-dependent.* Zooming into smallest scales the motions of particles in space should be even more *content-dependent.* This makes it difficult to impossible to find a definition of space and time.

Motion and State Implementation seem to be more fundamental than *Time.* Maybe the underlying categories could also be composite, leading to more open questions. *Euclidian Geometry* is omnipresent in our daily lives and appears to us as a perfect match between a theory and day-to-day physical reality. Although it fits well with our Meso World scales, we know it is incompatible with large scales, and not applicable to small scales.

On smallest Scales $(5, 3, 4)$ and $\frac{4}{3}$ might determine the speed of light. The triple $(5, 3, 4)$ defines the smallest possible values for implementing and relocating klicks. We related its multiples to conceptual and indirectly to real particles travelling at maximum speed like photons, which led to a speed limit determined by the relation of relocating to implementing klicks as $\mathfrak{T}[\frac{4}{3}, \frac{4}{3}]$. The triple $(5, 3, 4)$ defines an *end-point* in the model for speed/mass relations on smallest scales.

Can the model explain the same speed of photons having no mass and neutrinos with some mass? Maybe yes. The Klick Model defines the speed of any object as limited by the Klick Relation K_r/K_i not exceeding $4/3$. In other words, the maximum speed of a physical object is not determined by its mass related to K_i but through its relation between relocating and implementing klicks.

Probability of Quantum Theory [8] is possibly based on *Ambiguity* on smallest Scales. The Klick Model relies on intrinsic ambiguities of Pythagorean Triples and compatibility constraints of two-particle events. The bargain could be to exchange *probability* by *ambiguity.* If true, Quantum Theory would be an *approximation* to a finer-grained reality.

Acknowledgments

The idea to write up own ideas from the perspective of Informatics was triggered by an essay from astrophysicist Andreas Müller [13] explaning *time* as bound to our *mesoscopic* scales.[27]

[27] An extract of Andreas Müller's essay 'Was ist Zeit?' or 'What is Time?' [14]: "Simply said, the problem to understand the term *time* is a problem with scales: humans are living on a mesoscopic scale, which is the daily experienceable world. Quantum Theory governs microscopic scales. On this scale, there are processes difficult to understand like the tunnel effect, the uncertainty principle, the quantum vacuum, and manipulative measurements. On the other side, on a macroscopic scale, there are the laws of Relativity Theory. Also, here we find on a mesoscopic scale strange effects like time dilatation, Lorentz Contraction, curvature of space, and gravitational lenses. Even more, there are overlapping areas of the grand theories ...

These considerations should make clear that the classical physical theories [...] are on the mesoscopic scale: They are simply more intuitive because they can be transferred more easily to our today's world.

For a long time, until the beginning of the 20th century, the mesoscopic standpoint and the 'contemporary perspective' hampered the cognitive and scientific view ...

But even a hundred years after the big grand theories, Relativity and Quantum Theories, mankind still fights for an understanding. Old definitions are evolving into new. A main problem certainly exists because human terms stemming from mesoscopic thinking cannot be simply transferred to other scales. This is why it should not be a surprise that the understanding of the term *time* only make sense where it was invented: Our earth, moving in a slow and flat sector defined by Relativity Theory."

References

[1] Milne, R.E.;Strachey, C. (1976). A theory of programming language semantics. ISBN 978-1-5041-2833-9

[2] Price, H. Lee (2008). "The Pythagorean Tree: A New Species". https://arxiv.org/abs/0809.4324

[3] Lehmer, D. N. "Asymptotic Evaluation of Certain Totient Sums." Amer. J. Math. 22, 293-335, 1900

[4] C.B. Parker (1994). McGraw-Hill Encyclopaedia of Physics (2nd ed.). McGraw-Hill. pp. 1192, 1193. ISBN 0-07-051400-3.

[5] Thomas Lacroix et al., Predicting the dark matter velocity distribution in galactic structures, arXiv:2005.03955v2 [astro-ph.GA] 19 Oct 2020

[6] Weyl, Hermann, (1949), Philosophy of Mathematics and Natural Sciences, Princeton University Press, Princeton

[7] Einstein, A.: Zur Quantentheorie der Strahlung. In: Physik.Zeitschr. 18 (1917), S. 121-128

[8] Feynman, Richard P. (1985). QED: The Strange Theory of Light and Matter. Princeton University Press. ISBN 0-691-02417-0.

[9] A101929 - Number of Pythagorean triples with hypotenuse $< 10^n$, https://oeis.org/A101929

[10] Nadir Murru, Marco Abrate, Stefano Barbero and Umberto Cerruti, Groups and monoids of Pythagorean triples connected to conics, Open Mathematics,2017

[11] Hans de Vries, An exact formula for the Electro Magnetic coupling constant. http://chip-architect.com/news/2004_10_04_The_Electro_Magnetic_coupling_constant.html, October 4, 2004.

[12] On a recursive construction of circular paths and the search for π on the integer lattice $\mathbb{Z}^2$ Michelle Rudolph-Lilith, https://arxiv.org/abs/1602.06239 [cs.GR]

[13] https://de.wikipedia.org/wiki/Andreas_M%C3%BCller_(Astronom)

[14] Müller, Andreas, Was ist Zeit?, August 2007, Webessay, https://www.spektrum.de/astrowissen/zeit.html

Appendix A Tables of Klick Relations between CF- and VF-Particles

A.1 Tables for Pythagorean Triples and CF-/VF-Particles

Table 3 and Table 4 refer to the sets $\mathcal{P}]0,\infty[$ and $\mathcal{T}]0,\infty[$, Pythagorean Triples, which are not directly related to CF-/VF-Particles. Table 5 and Table 6 are limited by their quotient of b/a-legs in $]1,\frac{4}{3}[$ and therefore are meant to refer to CF-/VF-Particles.

Each line refers to triples (c,a,b) with $c^2 = a^2 + b^2$ for $c < M$ with $a < b$. The value N is the total number of triples with $c < M$. The values for $avgA$ and $avgB$ are determined correspondingly. Each value of c may refer to multiple triples, like $c = 65$, or to no triples, like $c = 1$.

The columns with $\overline{\pi}$ refer to a discrete version of π, borrowed from the work of Michelle Rudolph-Lilith [12]. I think the value is possibly the same for Table 3 as suggested in the article.

Table 3: Primitive Pythagorean Triples $\in \mathcal{P}]0,\infty[$

M	N	avgC	avgA	avgB	avgAB	avgA/avgB	$\overline{\pi}_{\mathcal{P}]0,\infty[}$	$\pi_{\mathcal{P}]0,\infty[}$
$1*10^5$	15918	50018	18653	45032	31843	0.4142271	3.1738169	3.1414320
$2*10^5$	31818	99969	37275	90005	63640	0.4141458	3.1739784	3.1415530
$4*10^5$	63668	200026	74577	180095	127336	0.4141010	3.1740456	3.1416025
$8*10^5$	127325	400011	149179	360135	254657	0.4142318	3.1740026	3.1415528
$16*10^5$	254648	800009	298364	720253	509309	0.4142488	3.1740119	3.1415560
$32*10^5$	509275	1599945	596669	1440454	1018561	0.4142228	3.1740397	3.1415782
$64*10^5$	1018585	3199991	1193333	2881015	2037174	0.4142058	3.1740565	3.1415915
$128*10^5$	2037175	6399977	2386668	5762016	4074342	0.4142071	3.1740609	3.1415943
$256*10^5$	4074385	12800062	4773419	11524111	8148765	0.4142115	3.1740603	3.1415931
$512*10^5$	8148717	25599970	9546819	23048071	16297445	0.4142134	3.1740595	3.1415923
$1024*10^5$	16297459	51199987	19093679	46096178	32594928	0.4142139	3.1740595	3.1415922
$2048*10^5$	32594898	102399893	38187262	92192315	65189789	0.4142131	3.1740599	3.1415927

Table 4: Primitive and non-primitive Pythagorean Triples $\in \mathcal{T}]0,\infty[$

M	N	avgC	avgA	avgB	avgAB	avgA/avgB	$\overline{\pi}_{\mathcal{T}]0,\infty[}$	$\pi_{\mathcal{T}]0,\infty[}$
$1*10^5$	161430	52468	21381	46598	33990	0.4588434	3.1099128	3.0846639
$2*10^5$	344883	104609	42409	92984	67696	0.4560861	3.1139976	3.0882300
$4*10^5$	733926	208672	84206	185620	134913	0.4536501	3.1175963	3.0913785
$8*10^5$	1556102	416358	167321	370608	268964	0.4514768	3.1207968	3.0941824
$16*10^5$	3288692	830956	332695	740084	536390	0.4495369	3.1236507	3.0966864
$32*10^5$	6930510	1658783	661889	1478173	1070031	0.4477751	3.1262244	3.0989480
$64*10^5$	14567048	3311866	1317452	2952694	2135073	0.4461865	3.1285435	3.1009888
$128*10^5$	30546276	6613411	2623443	5898781	4261112	0.4447433	3.1306466	3.1028419
$256*10^5$	63916820	13207988	5225969	11785501	8505735	0.4434236	3.1325656	3.1045345
$512*10^5$	133481757	26381420	10413628	23548851	16981240	0.4422138	3.1343213	3.1060848
$1024*10^5$	278260068	52699394	20756895	47057039	33906967	0.4411007	3.1359343	3.1075104
$2048*10^5$	579112771	105281717	41384016	94039027	67711522	0.4400728	3.1374213	3.1088259

Table 5: Primitive Pythagorean Triples $\in \mathcal{P}]1,\frac{4}{3}[$

M	N	avgC	avgA	avgB	avgAB	avgA/avgB	$\overline{\pi}_{\mathcal{P}]1,\frac{4}{3}[}$	$\pi_{\mathcal{P}]1,\frac{4}{3}[}$
$1*10^5$	2877	50046	32762	37776	35269	0.8672699	2.8379421	2.8379166
$2*10^5$	5746	99934	65428	75427	70428	0.8674338	2.8379440	2.8379185
$4*10^5$	11499	199972	130904	150949	140927	0.8672046	2.8379640	2.8379384
$8*10^5$	23003	399992	261861	301918	281889	0.8673251	2.8379607	2.8379350
$16*10^5$	46010	800062	523768	603898	563833	0.8673123	2.8379634	2.8379377
$32*10^5$	92009	1599949	1047418	1207668	1127543	0.8673064	2.8379648	2.8379391
$64*10^5$	184017	3199839	2094773	2415311	2255042	0.8672893	2.8379664	2.8379406
$128*10^5$	368053	6399948	4189725	4830823	4510274	0.8672901	2.8379667	2.8379410
$256*10^5$	736113	12800017	8379564	9661709	9020636	0.8672962	2.8379666	2.8379409
$512*10^5$	1472219	25599968	16759081	19323371	18041226	0.8672959	2.8379667	2.8379410
$1024*10^5$	2944448	51200095	33518233	38646891	36082562	0.8672944	2.8379668	2.8379411
$2048*10^5$	5888877	102399711	67036223	77293360	72164792	0.8672960	2.8379667	2.8379410

Table 6: Primitive and non-primitive Pythagorean Triples $\in \mathcal{T}]1,\frac{4}{3}[$

M	N	avgC	avgA	avgB	avgAB	avgA/avgB	$\overline{\pi}_{\mathcal{T}]1,\frac{4}{3}[}$	$\pi_{\mathcal{T}]1,\frac{4}{3}[}$
$1*10^5$	43997	51645	32746	39858	36302	0.8215725	2.8456068	2.8455572
$2*10^5$	91959	103123	65476	79512	72494	0.8234677	2.8452752	2.8452258
$4*10^5$	191898	205991	130955	158692	144823	0.8252098	2.8449708	2.8449216
$8*10^5$	399746	411505	261909	316768	289339	0.8268170	2.8446908	2.8446419
$16*10^5$	831385	822129	523816	632399	578107	0.8282995	2.8444328	2.8443842
$32*10^5$	1726559	1642632	1047629	1262696	1155162	0.8296770	2.8441939	2.8441456
$64*10^5$	3580684	3282230	2095246	2521478	2308362	0.8309594	2.8439721	2.8439242
$128*10^5$	7416491	6558805	4190467	5035663	4613065	0.8321580	2.8437656	2.8437180
$256*10^5$	15343251	13107072	8380904	10057723	9219313	0.8332804	2.8435728	2.8435255
$512*10^5$	31706905	26194329	16761684	20089920	18425802	0.8343331	2.8433924	2.8433455
$1024*10^5$	65454754	52351603	33523217	40132041	36827629	0.8353230	2.8432232	2.8431766
$2048*10^5$	134991369	104633546	67046123	80174247	73610185	0.8362551	2.8430642	2.8430180

The results are calculated based on a 64bit arithmetic up to a common maximum limit to avoid arithmetic overflows for some of the parameter combinations.

A.2 Table Columns and their Correlations

An explanation of columns in Table 3 is given below, including correlations between them.

$$avgC(M) = \frac{\sum\limits_{n=1}^{n<M} c}{n},\ avgA(M) = \frac{\sum\limits_{n=1}^{n<M} a}{n},\ avgB(M) = \frac{\sum\limits_{n=1}^{n<M} b}{n} \tag{A.1}$$

$$avgAB(M) = \frac{\sum\limits_{n=1}^{n<M} \frac{a+b}{2}}{n} \tag{A.2}$$

$$avgAbyAvgB(M) = \frac{avgA(M)}{avgB(M)} \tag{A.3}$$

$$\overline{\pi}_{\mathcal{P}]0,\infty[}(M) = 2 * \frac{\sum\limits_{n=1}^{n<M} \frac{c}{avg(a,b)}}{n} \tag{A.4}$$

$$\pi_{\mathcal{P}]0,\infty[}(M) = 2 * \frac{n}{\sum\limits_{n=1}^{n<M} \frac{avg(a,b)}{c}} \tag{A.5}$$

Based on this table, increasing values of $maxC$ seem to converge to the following values[28].

$$M/N \rightarrow \frac{1}{2\pi} \tag{A.6}$$

$$M/avgAB \rightarrow \frac{1}{\pi} \tag{A.7}$$

$$avgA/avgB \rightarrow \frac{1}{1+\sqrt{2}} \tag{A.8}$$

Because we want to associate motion/state propagation of real world particles with a subset of $\mathcal{T}]0,\infty[$, the above tables are not really helpful. The set $\mathcal{P}]0,\infty[$ looks promising because of the above convergences, but it is unclear how to come from $\mathcal{P}]0,\infty[$ to $\mathcal{T}]0,\infty[$.

[28] For the first converging values of A.6, see Lehmer [3].

A.3 Algorithm

```java
// Program generating the Tables for Klick Model Appendix A (V1.5)
import java.io.PrintWriter;
public class GenAppendixA {
  // ---------------------- Parameters & Counters ----------------------
  static long    M      = 2048*100000; // maximum allowed "c-value" for a Pythagorean Triple ("radius of C-Values")
  static boolean  withMultiples = false;  // true, if non-primitive triples are included
  static boolean  with3x4Limits = false;   // true, if only triples in T]1,4/3[ included
  static boolean  printTriples  = false;   // true, if triples printed to text file
  static long     nTriples      = 0;     // number of enumerated triples
  static long     sumC=0, sumA=0, sumB = 0;  // sum over all c,a,b-values of all enumerated triples
  static double   sum_avg_ab    = 0.0;     // sum over all avg(a,b) - values
  static double   sum_c_by_avg_ab = 0.0;   // sum over all (c/avg(a,b) - values
  static double   sum_avg_ab_by_c = 0.0;   // sum over all (avg(a,b)/c) - values
  static PrintWriter pw = (PrintWriter)null;   // output files for results

  public static void main(String[] args)
  {
    long startTime = System.currentTimeMillis();
    generateTriples( "AppendixA.txt" );
    System.out.println( "Done in " + (System.currentTimeMillis() - startTime)/1000 + " sec." );
  }

  static void generateTriples( String filename )
  {
    try {
      pw = new PrintWriter( filename, "UTF-8" ); // output file for results
      for ( int table = 1; table <= 4; ++ table ) // run through 4 parameter settings to create tables
      {
        if ( table == 1 ) { withMultiples = false; with3x4Limits = false; } else
        if ( table == 2 ) { withMultiples = true; with3x4Limits = false; } else
        if ( table == 3 ) { withMultiples = false; with3x4Limits = true; } else { withMultiples = true; with3x4Limits = true; }

        System.out.println( "withMultiples=" + withMultiples + " and with3x4Limits=" + with3x4Limits
                  + "\nM & N & avg-c & avg-a & avg-b & avg-ab-by-n & avg-a-by-avg-b"
                      + "& sum-c-by-avg-ab-by-n-times-2 & n-by-sum-avg-ab-by-c-times-2\\\\" );

        for ( long maxC = 100000; maxC <= M; maxC *=2 ) // more produces arith. overflows for some combinations with a 64bit arithmetic)
        {
          nTriples = 0; sumC = sumA = sumB = 0; sum_avg_ab = sum_c_by_avg_ab = sum_avg_ab_by_c = 0.0;

          enumTriples( maxC, 3, 4, 5 );

          double avg_c = (double)sumC / nTriples;
          double avg_a = (double)sumA / nTriples;
          double avg_b = (double)sumB / nTriples;
          double avg_ab_by_n = sum_avg_ab / nTriples;
          double avg_a_by_avg_b = avg_a / avg_b;
          double sum_c_by_avg_ab_by_n_times_2 = 2 * (sum_c_by_avg_ab / nTriples);
          double n_by_sum_avg_ab_by_c_times_2 = 2 * (nTriples / sum_avg_ab_by_c);

          String maxCStr = (maxC/100000) + "*10^5";
          String str = String.format( "$%s$ & %d & %.0f & %.0f & %.0f & %.0f & %.07f & %.07f & %.07f\\\\", maxCStr, nTriples,
                avg_c, avg_a, avg_b, avg_ab_by_n, avg_a_by_avg_b, sum_c_by_avg_ab_by_n_times_2, n_by_sum_avg_ab_by_c_times_2 );
          String str2 = String.format( "%d\t%d\t%.0f\t%.0f\t%.0f\t%.0f\t%.010f\t%.012f\t%.012f", maxC, nTriples,
                avg_c, avg_a, avg_b, avg_ab_by_n, avg_a_by_avg_b, sum_c_by_avg_ab_by_n_times_2, n_by_sum_avg_ab_by_c_times_2 );
          System.out.println( str ); pw.println(str2);
        }
      }
      pw.flush(); pw.close();
    } catch (Exception e) { e.printStackTrace(); System.exit(1); }
  }

  // Enumerate Pythagorean Triples with one of the ternary tree methods (see http://en.wikipedia.org/wiki/Tree_of_primitive_Pythagorean_triples).
  public static void enumTriples ( long maxC, long a, long b, long c )
  {
    if ( c >= maxC || a >= maxC || b >= maxC ) return;
    addTriple( maxC, a, b, c );
    enumTriples ( maxC, a - 2*b + 2*c, 2*a - b + 2*c, 2*a - 2*b + 3*c );
    enumTriples ( maxC, a + 2*b + 2*c, 2*a + b + 2*c, 2*a + 2*b + 3*c );
    enumTriples ( maxC, -a + 2*b + 2*c, -2*a + b + 2*c, -2*a + 2*b + 3*c );
  }

  // evaluate new primitive triple and its multiples depending on parameter setting
  static void addTriple ( long maxC, long a, long b, long c)
  {
    if ( a > b ) { long x = a; a = b; b = x; } // make sure a < b
    evalTriple( a, b, c );

    if ( withMultiples ) // add all multiples of a primitive triple
      for ( long j = 2; c*j < maxC; ++j )
        evalTriple( a*j, b*j, c*j );
  }

  // evaluate triple - always being invoked with a < b
  static void evalTriple( long a, long b, long c )
  {
    if ( (with3x4Limits && ( b*3 > a*4 )) || c == 5 ) return; // exclude also (5,3,4)-Triple, not being included in T]1,4/3[
    ++nTriples;
    if ( printTriples ) pw.println( c + "\t" + a + "\t" + b );
    double avg_ab = (a+b)/2.0; sumC += c; sumA += a; sumB += b; sum_avg_ab += avg_ab; sum_c_by_avg_ab += c/avg_ab; sum_avg_ab_by_c += avg_ab/c;
  }
}
```

Appendix B Algorithm enumerating Generic Ensembles

```java
// Source file: GenericEnsembles.java - Generating Generic Ensembles / V1.5a 27.01.2025
import java.io.PrintWriter;

public class BlackBody {
  // ---------------------- Parameters & Counter -----------------------
  static long     MaxC       = 800000;    // maximum M
  static boolean    withMultiples  = false;  // true, if non-primitive triples are included (not used in the paper)
  static boolean    withRatioLimit  = true;    // true, if only triples in T[4/3,1] included (true for use in paper)
  static boolean    withPrinting   = false;   // true, if output to file Statistics.txt
  static boolean    withPrEStatesStat = false;   // true, if to print number of states per level
  static boolean    withLimitLevel2One = false;  // limit output to two levels, so that nLevel = 1

  public static void main(String[] args)
  {
    long startTime = System.currentTimeMillis();

    String filename = "Statistics.txt";

    try {
      PrintWriter pw = new PrintWriter( filename, "UTF-8" );
      System.out.println( "Writing to: " + filename );

      enumTriples( 3, 4, 5 );

      Triple.addExcStates( MaxC, withLimitLevel2One, pw, withPrinting );
      if ( withPrinting ) Triple.printTriples( pw );

      System.out.println("Evaluating results after " + (System.currentTimeMillis() - startTime)/1000 + " sec. ...");
      Triple.evalResults( MaxC, pw, withPrEStatesStat );

      pw.flush(); pw.close();
    } catch (Exception e) { e.printStackTrace(); System.exit(1); }

    System.out.println( "Done checking " + Triple.nTriples + " triples in " + (System.currentTimeMillis() - startTime)/1000 + " sec." );
  }

  /* ----------------------- Generate Pythagorean Triples ----------------------------- */
  public static void enumTriples ( long a, long b, long c )
  {
    if ( c >= MaxC ) return;
    addTriple( MaxC, a, b, c );

    enumTriples ( a - 2*b + 2*c, 2*a - b + 2*c, 2*a - 2*b + 3*c );
    enumTriples ( a + 2*b + 2*c, 2*a + b + 2*c, 2*a + 2*b + 3*c );
    enumTriples ( -a + 2*b + 2*c, -2*a + b + 2*c, -2*a + 2*b + 3*c );
  }

  // evaluate new primitive triple and its multiples depending on parameter setting
  static void addTriple ( long maxC, long a, long b, long c)
  {
    if ( a > b ) { long x = a; a = b; b = x; } // make sure a < b
    if ( withRatioLimit && (b*3 > a*4) || c == 5 ) return;

    Triple.mkTriple( a, b, c );

    if ( withMultiples ) // add all multiples of a primitive triple
      for ( long j = 2; c*j < MaxC; ++j )
        Triple.mkTriple( a*j, b*j, c*j );
  }
}
```

```java
// Source file: Triple.java - Generating Generic Ensembles / V1.5a 27.01.2025
import java.io.PrintWriter;

public class Triple {
  long         a, b, c;   // with c^2 = a^2 + b^2
  State        s;       // list of states, i.e. combinations of ground states with multiples of (5,3,4)
  int          nStates;   // length of this list
  double         p;       // quotient between successif values of total c-values (s.tc)

  static final long  N_TRIPLES  = 100000000; // should be more than needed for an example
  static Triple    triples[]  = new Triple[(int)N_TRIPLES]; // stored triples
  static int       nTriples   = 0;   // number of VF-Particles in isolated or ground states
  static int       nEStates   = 0;   // number of VF-Particles in excited state
  static int       nIStates   = 0;   // number of VF-Particles in ground states not having exited states in the set
  static long      nGStates   = 0;   // number of VF-Particles in ground states having at least one excited state
  static long      sumOfFs    = 0;
  static long      sumOffPkgs  = 0;
  static int       fPkgMax    = 0;   // largest fPkg value in the set of states
  static int       nLevel     = 0;

  // enter a triple into the triples array triples[]
  public static Triple mkTriple( long a, long b, long c ) // always being invoked with a < b
  {
    Triple newt = new Triple();
    newt.a = a; newt.b = b; newt.c = c;

    if ( nTriples >= N_TRIPLES )
    {
      System.err.println( "Error: triples[] array overflow -> increase N_TRIPLES" ); System.exit(1);;
    }
    triples[nTriples++] = newt;
    return newt;
  }
```

```
static boolean isTriple( long a, long b )
{
  long cc = a*a + b*b;
  long c = (long)Math.sqrt(cc);
  if ( c*c != cc ) return false;      // not a perfect square
  return true;
}

// evaluate, which CF-States with factor "f" can be added to a ground state forming an excited state
public static void addExcStates( long maxC, boolean withLimitLevel2One, PrintWriter pw, boolean withPrinting )
{
  // for ground states, which are described by PPTs ...
  for ( int i = 0; i < nTriples; ++i )
  {
    Triple t = triples[i];
    boolean hasExcStates = false;
    State lastState = null;

    for ( int f = 1, count = 0; ; ++f )
    {
      long tt_a = 3*f, tt_b = 4*f;
      long a = t.a + tt_a, b = t.b + tt_b;
      if ( a <= 0 || b <= 0 ) { System.out.println("Error -> Arithmetic Overflow!"); continue; }

      long cc = a*a + b*b;
      long c = (long)Math.sqrt(cc);
      if ( c >= maxC ) break;  // limit while adding 534 multiples
      if ( c*c != cc ) continue; // not a perfect square

      hasExcStates = true;
      State s = new State(); ++nEStates;
      s.c = c; s.a = a; s.b = b; s.tb = (t.c + 4*f);
      s.f = f; s.level = count++;
      s.fPkg = t.s == null ? f : f - lastState.f;
      s.p = lastState == null ? (s.tb-t.b)/(double)s.b : (s.tb-lastState.tb)/(double)lastState.tb;
      sumOfFs += f; sumOffPkgs += s.fPkg;

      if ( t.s == null ) { t.s = s; lastState = s; } else lastState.next = s;
      lastState = s;

      if ( s.fPkg > fPkgMax ) fPkgMax = (int)s.fPkg;
      if ( count > nLevel ) { nLevel = count; System.out.println("nLevel=" + nLevel); }
      if ( withLimitLevel2One ) break;
    }
    if ( !hasExcStates ) ++nIStates;
    t.p = ((double) t.nStates) / ((maxC-t.c)/5);
  }
  nGStates = nTriples - nIStates;
}

public static void printTriples( PrintWriter pw )
{
  for ( int i = 0; i < nTriples; ++i )
  {
    Triple t = triples[i];
    if ( t.s == null ) continue; // print only excited states

    String line = "", tripleStr = t.c + " " + t.a + " " + t.b /** " " + String.format("%.09f", t.p)*/;

    for ( State s = t.s; s != null; s = s.next )
    {
      line += " " + s.tb; // line += " " + s.fPkg; //line += " " + String.format("%.09f", s.p);
    }
    pw.println( tripleStr + " " + line );
  }

  // eval excited states per level
  long nPerLevel[] = new long[(int)nLevel];
  double pPerLevel[] = new double[(int)nLevel];

  for ( int i = 0; i < nTriples; ++i ) // count all fDeltas ...
  {
    Triple t = triples[i];

    for ( State s = t.s; s != null; s = s.next )
    {
      ++nPerLevel[s.level]; pPerLevel[s.level] += s.p;
    }
  }

  String l1 = "x x " + nTriples + " ", l2 = "x x x ";
  for ( int i = 0; i < nLevel; ++i )
  {
    l1 += " " + nPerLevel[i]; l2 += " " + String.format("%.05f", pPerLevel[i] / nPerLevel[i]);
  }
  pw.println( "---" + "\n" + l1 + "\n" + l2 );
}
```

```java
  public static void evalResults( long maxC, PrintWriter pw, boolean printEStatesStat )
  {
    long nB12 = 0L, nB21 = 0L, nA21 = 0L, nUniqPkg = 0L;
    int count[] = new int[fPkgMax+1];
    boolean done[] = new boolean[fPkgMax+1];

    // count all fPkg's ...
    for ( int i = 0; i < nTriples; ++i )
    {
      Triple t = triples[i];
      for ( State s = t.s; s != null; s = s.next )
        ++count[(int)s.fPkg];
    }

    // calculate g_* values
    int gMax = 0;
    for ( int i = 0; i < count.length; ++i )
    {
      int f = count[i];
      if ( f == 0 ) continue;
      if ( f > gMax ) gMax = f;
    }

    for ( int i = 0; i < nTriples; ++i ) // for each ground state ...
    {
      Triple t = triples[i];
      if ( t.s == null ) continue;

      for ( State s = t.s; s != null; s = s.next )
      {
        int this_fPkg = (int)s.fPkg;
        int fPkgCount = count[this_fPkg];

        if ( done[this_fPkg] ) continue;  // already processed at first occurence of this value
        done[this_fPkg] = true;    // avoids to process this fPkg value for its next instances
        ++nUniqPkg;           // add to number of unique excited states

        if ( fPkgCount == 1 )
        {
          ++nB12; ++nA21; continue;  // only one occurance of this fPkg value
        }

        // otherwise there are multiple instances of this fPkg value
        nB12 += fPkgCount; nB21 += fPkgCount;
      }
    }

    if ( printEStatesStat )
    {
      int cnt[] = new int[gMax+1], cntSum[] = new int[gMax+1];
      done = new boolean[fPkgMax+1];

      // for all excited states ...
      for ( int i = 0; i < fPkgMax; ++i )
      {
        if ( count[i] == 0 || done[i] ) continue;
        pw.println( i + "\t" + count[i] );
        ++cnt[count[i]]; cntSum[count[i]] += count[i];
        done[i] = true;
      }
      pw.println( "---" );
      for ( int i = 0; i < gMax+1; ++i )
      {
        if ( cnt[i] > 0 )
          pw.println( i + "\t" + cnt[i] + "\t" + cntSum[i] );
      }
    }

    String line = "MaxC=" + maxC + " nTriples=" + nTriples + " nIStates=" + nIStates + " nGStates=" + nGStates + " nEStates=" + nEStates + "
          nUniqPkg=" + nUniqPkg
          + " nA21=" + nA21 + " nB21=" + nB21 + " nB12=" + nB12
          + " fAvg=" + sumOfFs / nEStates + " fPkgAvg=" + sumOffPkgs / nEStates
          + " nLevel=" + nLevel + " gMax=" + gMax
          + " r_n=" + String.format("%.09f", ((double)nA21 * nEStates) / (nB12 * (nTriples-nIStates) - nB21 * nEStates));
    System.out.println( line );
  }
}
```

```java
// Source file: State.java - Generating Generic Ensembles / V1.5a 27.01.2025
public class State {
  long a,b,c,tb; // overall values including factors of (5,3,4)-Multiples and total c-value with f*5
  long f;      // factor - a multiple of (5,3,4) to be added to the ground state
  long fPkg;  // multiple of (5,3,4) appied to the underlying ground or excited state
  int  level;   // position in chain of states
  double p;      // to calculate probability to get next f
  State next;  // next state for a given groundstate plus f*(5,3,4)
}
```