Relativistic Time Dilation & Information Science

Serge Mijatovic
PO Box 731, Murrieta, CA 92564
e-mail sergemij@yahoo.com

The effect of apparent 'slowing' of time due to motion and gravity can be explained in terms much simpler than those of Special Relativity (SR) and General relativity (GR). Instead of involving the speed of light, gravity, or counter-intuitive premises, postulates or principles, this effect can be derived by using simple deductions based on information science. The formulas developed here reduce to SR and GR in limiting cases.

1. Introduction

In relativity theories, a perception of 'slowing' is called 'time dilation', implying that time itself slows down. But here, the perception of slowing will not be called 'time dilation', because we consider time itself to be invariant. Instead, we suppose it is some internal mechanism of all physical processes that slows down, making all physical processes slower. We will call this effect 'performance-hit', to differentiate it from the traditional notion of time dilation *per se*.

We could say that an ordinary computer runs slower due to increased information load, which is easy to understand as a 'performance-hit'. We could then generalize that if information usage is at the core of reality itself, the performance-hit is a consequence of it. So the time itself cannot dilate, and only the throughput of information usage changes, causing the rate of physical processes to vary. If we are able to produce the same results as relativity without the notions of time dilation and space-time, we ought to start with a simpler setup, which is that of a flat 3D Euclidean space and not a 4-D Minkowski space.

This approach does not require any other postulates or principles to begin with. This is a rather big difference to Einstein's relativity theories, SRT and GRT. But we can nevertheless derive (as limiting cases) not only Einstein's formulas for time dilation, but also his postulate about the speed of light.

In addition, we will deduce the concept of mass and show a simple derivation of the Heisenberg's Uncertainty Principle and Newton's Law of Gravitation starting only from the same axiomatic principles of information science. This allows us to claim a broader framework for this idea as it unites concepts that were so far only known as disjoined postulates and principles.

In limiting cases, the formulas developed reduce exactly to the results of SRT and GRT. Experimentally verifiable results such as with particle accelerators or GPS are the same. Sometimes interpretations of results are different. For example, the underlying cause of performance-hit is the same regardless of whether it arises out of relative movement or from the presence of other masses.

We introduce here the axiom of information use:

Any physical effect occurs only due to possession and use of information.

An aspect of this axiom that is applicable to fundamental physics is clearly this: an effect that may be attributable to any actionable concepts of physics (be it fields, particles, forces, probabilities, *etc*.), can come only because there is information to affect it. We call it 'axiom' because it is axiomatic that an action has a reason and a reason is based on information possession and use.

To facilitate development of this idea, we will adopt terms of 'information processing' or 'computation' (denoting use of information), and a 'computer' (a fundamental physical setup that allows use and possession of information). Those terms are not to be confused with ordinary computers, nor their modes of processing information. However, the same basic laws of information science apply to both, just as the axiom of information use does. For this reason, we will start with an analogy to ordinary computers.

Imagine that a computer is running a clock application that shows clicks of time. Now start typing. In fact type so fast, that the processing power a computer uses starts to take it away from the clock. What will you see? The clock will slow down.

Now imagine that a computer in your car is calculating your speed and your position. Each task uses some memory. As you type faster at the computer, the text you type is using more and more memory. So, less and less memory is used for speed and position calculations. If you want to use more memory for speed, there is less memory for position, and *vice versa*. If you want your speed calculated to a high degree, the direction in which you're going will be uncertain, and *vice versa*.

Keep these simple analogies in mind as you continue reading.

2. Information

We start from the least possible informational model that abstains from any specific representation or method of information use.

A fact is an elementary description of something, a description that cannot be simplified. Information (or 'data') is a collection of facts. All facts have equal significance.

A computer can store and compute information. A computer has fixed information storage and fixed throughput of processing information. A computer is deterministic, *i.e*. the same input always produces the same output. If there is more information than storage available, then information is processed to fit (we call this processing a 'compression'). A computer must not be stateless or otherwise the behavior would never depend on what came before.

A computer has a definitive location and occupies space around it where information is stored and computing takes place. The space where the computer stores information has no preference for scale or direction. Any sphere centered at a computer has all of the computer's information. So the information at some distance R from a computer is:

$$i_R = a \times i / R^2 \quad (1)$$

where i_R is the average number of facts at distance R from the computer over some fixed small surface area a; R is the distance from the computer; i is the information-storage of the computer. For simplicity, we henceforth omit the a.

Processing of information means combining facts to produce new facts. In the case of two sets of facts, one with N facts and the other with M facts, the cost of computing a new set of facts combining the two is proportional to $N \times M$.

2.1 Usage of Information

Available information at some location is the sum of all information from all computers at that location.

A computer uses available information and creates new information. Since available information can generally be much larger than the computer's storage, the data has a random subset of the available-information (see Figure 1).

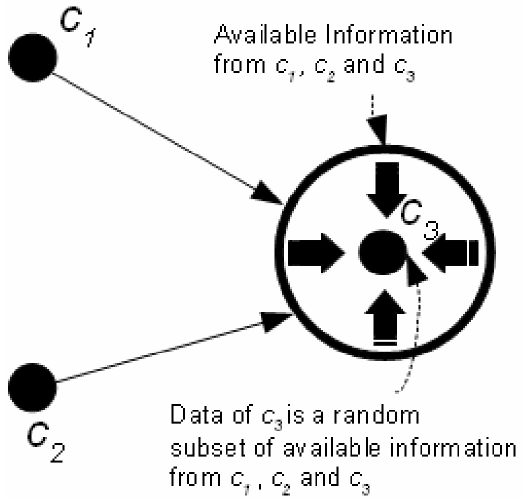

Figure 1.

In order to make the most accurate use of available-information, new facts are added to storage whenever the available information changes. A computer must have a 'memory' to account for the past. Without accounting for the past, a computer would be a stateless machine. In a simplest scenario, computer's storage is divided into 'previous' and 'current'. We will call them 'states' or 'sets'.

2.2 Distance Effect

The total available information at a location of a computer is:

$$i_t = i + i_a \quad (2)$$

i_t is the total available information, i is the computer's data, and i_a is the information of all other computers.

The proportion of available information with and without the computer itself (f) is:

$$f = i_a / i_t = i_a / (i + i_a) = 1/(1 + i / i_a) \quad (3)$$

i_a can be written as a sum:

$$i_a = \sum_{j=1}^{U} i_j \Big/ R_j^2 \quad (4)$$

where U is the number of all other computers; i_j is the information of the other computer j; R_j is the distance to computer j.

In general, a quantity f is describing the information-influence of the Universe:

$$f = 1 \Big/ \left\{ 1 + i \Big/ \sum_{j=1}^{U} i_j / R_j^2 \right\} \quad (5)$$

The information influence tells us how much computing capacity is used on other computers (*vs.* itself). The information influence of a computer X on any given computer is, from Eq. (3),

$$\begin{aligned} f_X &= \left(i_X / R_X^2\right) \Big/ \left[i + \left(i_X / R_X^2\right) + \sum_{j=1, j\neq X}^{U} \left(i_j / R_j^2\right) \right] \\ &= 1 \Big/ \left\{ 1 + \left[i / i_X \right] \times R_X^2 + \sum_{j=1, j\neq X}^{U} \left(i_j \big/ i_X\right) \times \left(R_X^2 \big/ R_j^2\right) \right\} \end{aligned} \quad (6)$$

Apparently:

$$0 < f_X < 1 \quad (7)$$

It is clear from Eq. (6) that

$$\sum_{1}^{U+1} f_j = 1 \quad (8)$$

If a computer is very far away from other computers, its own data is all that is available to it. So the available information from any computer is:

$$i \times f_X \quad (9)$$

If X is the only significant computer nearby, then from Eq. (6)

$$f_X \approx 1 \Big/ \left[1 + (i / i_X) \times R_X^2 \right] \quad (10)$$

If X is large and close, then from Eq. (10)

$$f_X \approx 1 \quad (11)$$

When X is small and far away:

$$f_X = 0 \quad (12)$$

The small computers are overwhelmed by the facts from nearby large computers (and *vice versa*).

The amount of information used from any computer X is always smaller than or equal to the data storage of X. This implies an inherent information loss in computing, even if computation itself is deterministic.

2.3 Relative-Motion Effect

If C moves relative to M, the number of locations visited in a given period of time will be proportional to its speed, and so will be the amount of additional information. A nearby computer Z at rest relative to M will not see this additional information.

The conclusion is that the relative motion has a real effect on information processing. Hence, the change of C's data is proportional to speed relative to M:

$$\Delta i / i = s \times v \times f_M \tag{13}$$

where Δi is the change in current set at C resulting from the movement relative to M; i is the size of C's current set; s is a dimensional constant of proportion; v is a relative speed achieved where f_M can be considered constant; f_M is the information-influence of M at C. It is a number between 0 and 1, effectively describing a proportion of computational resources used for processing information of M by C.

The exact equation for Δi would include all the computers in the Universe:

$$\Delta i = s \times i \times \sum_{j=1}^{U+1} v_j \times f_j \tag{14}$$

Note that Δi cannot become greater than i due to limited information storage:

$$\Delta i \le i \tag{15}$$

3. Information Management

3.1 Storage of Information

The aforementioned previous and current sets (that share the data storage) are computed together to create the new information. The previous set is the current set from the moment ago:

$$i_{\text{previous}}(t) = i_{\text{current}}(t - \Delta t) \tag{16}$$

Between the two, they can fill a fixed total storage capacity:

$$i_{\text{previous}} + i_{\text{current}} = \text{constant} \tag{17}$$

Δi denotes additional information to fit into storage:

$$(i_{\text{previous}} - \Delta i) + (i_{\text{current}} + \Delta i) = \text{constant} \tag{18}$$

The expression above shows a conceptual information flow of a computer. The previous set becomes compressed in order for current set to enlarge, when available information increases.

3.2 Processing Requirements

The cost of computing in the lossless case (when available information does not change; *i.e.* $\Delta i = 0$, a lossless case) is :

$$H_0 = i \times i = i^2 \tag{19}$$

In case of different amounts of information in previous and current sets; *i.e.* $\Delta i \ne 0$, a case of information loss, the cost is:

$$H = (i - \Delta i) \times (i + \Delta i) = i^2 - (\Delta i)^2 \tag{20}$$

Apparently,

$$H < H_0 \tag{21}$$

When $\Delta i > 0$, there is more information than there is storage. In this case, to minimize the loss of information, the previous set must be compressed to occupy less storage. This compression is denoted as a lossy transformation Q. The time spent for this compression is not used in computing the change – it only produces a lossy version of the same information, thus it can be considered 'not useful'. For the reasons outlined in this paragraph, the quantities H and H_0 will be referred to as 'useful'.

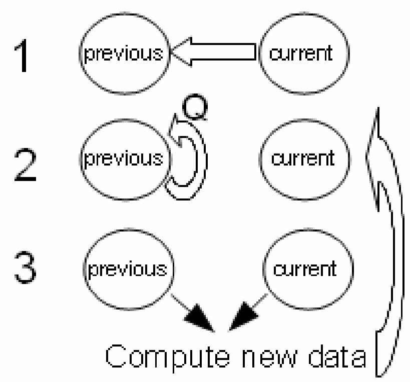

Figure 2.

The compression Q effectively computes information Δi with itself and so the cost is:

$$H_Q = (\Delta i) \times (\Delta i) = (\Delta i)^2 \tag{22}$$

The total cost is:

$$H_1 = H + H_Q = (i - \Delta i) \times (i + \Delta i) + (\Delta i)^2 = i^2 \tag{23}$$

And we have:

$$H_0 = H_1 \tag{24}$$

This result means that the total cost is the same both in lossless and lossy case. However in the case of information loss, the amount of useful computing is less; see Eq. (20).

3.3 Processing Throughput

The computational throughput T is the amount of useful information produced per unit of time. In the lossless case,

$$T_{\text{lossless}} = i / t(i) \tag{25}$$

where $t(i)$ is the time needed to process information i.

The throughput of computation for a lossy case can be deduced by modeling it with the equivalent situation where the two information sets are equal and both are lossless:

$$i_e \times i_e = (i - \Delta i) \times (i + \Delta i) \tag{26}$$

The useful throughput of processing information in the lossy case would be quantified with:

$$T_{\text{lossy}} = i_e / t(i_e) = \frac{\sqrt{(i - \Delta i) \times (i + \Delta i)}}{t(i)} = \frac{\sqrt{i^2 - (\Delta i)^2}}{t(i)} \tag{27}$$

or:

$$T_{\text{lossy}} = T_{\text{lossless}} \times \sqrt{1 - (\Delta i)^2 / i^2} \quad . \tag{28}$$

What T means is that, for example, if the size of both information sets together is 20, and the size of each set is 10 (so $10 + 10 = 20$), the amount of useful information processed would be a square root of 10×10, or 10 per unit of time. If the current set size is 11 and the previous size is 9 (so $11 + 9 = 20$), then the amount of useful information processed would be a square root of 11×9, or approximately 9.95 per unit of time. When the amount of available information increases, the lower throughput of useful information is the direct consequence of the limited information storage. When the amount of available information decreases, the throughput is higher, with the highest throughput achieved when a computer is far enough from other computers, in which case its own data is all that's available to it. In further text, the term 'information' will have a connotation of useful information.

4. Local Speed Limit

4.1 Speed Limit Near Large Computers

The dimensional constant s from Eq. (13) has a meaning beyond just being a constant of proportion. Let us consider what happens when speed v_c is such that additional data in current set fills the entire previous set:

$$\Delta i = i \tag{29}$$

This means that speed v is so high that the processing of information slows down to the point where it becomes zero (or near zero) because:

$$T_{\text{lossy}} = \frac{\sqrt{i^2 - (\Delta i)^2}}{t} = \frac{0}{t^2} = 0 \tag{30}$$

This speed then locally becomes the highest attainable relative speed of a computer resulting from its own computation.

From Eqs. (13) and (29) we have:

$$s = 1 / (v_C \times f_M) \tag{31}$$

In a system of two isolated computers M and C, where M is much larger then C:

$$f_M \approx 1 \tag{32}$$

We will denote this maximum local speed v_c simply as c. From above two equations:

$$s \approx 1 / c \tag{33}$$

It is clear that the value of c depends on the location. Near large computers, the value of c is practically a constant, as in Eq. (13).

However, farther from large computers, the following can be true:

$$f_M << 1 \tag{34}$$

In this case, the value of c can be much higher.

4.2 Speed Limit in the General Case

An exact value for maximum relative speeds c_j (relative to every other computer) in a given location [from Eqs. (14) and (29)] can be found by solving the following equation for every computer:

$$\sum_{j=1}^{U+1} c_j \times f_j = 1 / s \tag{35}$$

In different locations and for different relative speeds, the maximum speed c can be different, so:

$$c = B_c(r_j, v_j, i_j, \forall j) \tag{36}$$

r_j is the distances to all computers; v_j is the speeds relative to all computers; i_j is the amounts of data for each computer; B_c is the function representing the solution of equation in Eq. (35).

An expression for speed c for a given computer near a large isolated M is:

$$c = i / (s \times f_M) \tag{37}$$

As f_M can vary between 0 and 1, the speed c can vary too, depending on the location:

$$1 / s < c < \max(B_c) \tag{38}$$

The exact value for its maximum depends on the locale and the moment in time. If the information influence of other computers is sufficiently small, B_c can become arbitrarily big:

$$\lim_{f_j \to 0, \forall j} \left[\max(B_c) \right] \to \infty \tag{39}$$

Near large computers, speed c has its minimum value of $1/s$, while away from them it can be much higher, *i.e.* it is location-dependent.

To expand on this, imagine a computer that has attained a speed of:

$$c_1 = 1/s \tag{40}$$

in a location where that speed is a limit. Let that computer now move to a location where the limit is:

$$c_2 = 10/s = 10 \times c_1 \tag{41}$$

The computer will still have a speed of c_1 unless there is a reason to accelerate. Conversely if a computer has attained a speed of $10 \times c_1$, and then moves to a location where the limit is c_1, its relative speed to computers near that location will remain the same and be beyond the local speed limit.

Eqs. (17) and (18) imply that far from large computers, the computer C can accelerate to maximum speeds greater than those attainable near large computers. In addition, the larger the computer C is, the higher the speed limit for it will be.

A noteworthy consequence of Eq. (33) is that if a computer tries to obtain its own maximum speed near large computer C, it will always hit the same speed limit c relative to C, regardless of what was its initial speed.

5. Different Times

5.1 Application Time

Let us say there is an application running on a computer (representing some physical process). A computer has its own throughput of computation, which is the same always. An application though will work faster or slower, depending on the throughput of processing useful information. Let us formalize how much will an application slow down.

We will call the time measured by an application the 'application time'. The actual time for which computer will run will be the 'real time'.

The throughput of computation T can vary:

$$T_1(t) = \sqrt{i^2 - (\Delta i_1)^2} \Big/ t \quad , \quad T_2(t) = \sqrt{i^2 - (\Delta i_2)^2} \Big/ t \quad , \quad T_1(t) \neq T_2(t) \quad . \tag{42}$$

Let us now introduce t_1 and t_2 to be application-times measured by an application at two different moments in time.

The application throughput measured in terms of application-time must be the same:

$$T_1(t_1) = T_2(t_2) \tag{43}$$

We have

$$\sqrt{i^2 - (\Delta i_1)^2} \Big/ t_1 = \sqrt{i^2 - (\Delta i_2)^2} \Big/ t_2 \tag{44}$$

and

$$t_1 = t_2 \times \sqrt{i^2 - (\Delta i_1)^2} \Big/ \sqrt{i^2 - (\Delta i_2)^2} \tag{45}$$

From this and Eq. (14) we have

$$t_1 = t_2 \times \sqrt{\frac{1 - s^2 \times \left(\sum_{j=1}^{U} v_{j1} \times f_{j1} \right)^2}{1 - s^2 \times \left(\sum_{j=1}^{U} v_{j2} \times f_{j2} \right)^2}} \tag{46}$$

This represents the general transformation of application-time, where: t_1 is the application-time when computer speeds are v_{j1} relative to all other computers, and each such computer having information-influence of f_{j1}; t_2 is the application time with relative speeds of v_{j2} and information influences of f_{j2}.

The conclusion is that application-times of computers differ when in motion relative to other computers.

5.2 Limiting Cases

Let us consider a situation of a small moving computer C near large isolated computer M. The information-influence f_M is nearly 1, and the information-influence of all other computers is nearly zero.

Let us have t_1 and t_2 such that the two computers are at rest ($v_1 = 0$) for a unit of application-time t_1, and the relative speed is v_2 for a unit of application-time t_2:

$$f_M \approx 1 \quad , \quad f_j \approx 0, \forall j, \; j \neq M, \quad v_1 = 0, v_2 = v \neq 0 \tag{47}$$

For a small computer, with Eq. (33), Eq. (46) becomes

$$t_1 \approx t_2 \times \sqrt{\frac{1 - s^2 \times 0^2}{1 - s^2 \times (v+0)^2}} = \frac{t_2}{\sqrt{1 - s^2 \times (v+0)}} = \frac{t_2}{\sqrt{1 - v^2 / c^2}} \ . \tag{48}$$

Application time runs slower for a small computer C when moving at speed v (c is C's maximum speed attainable locally through own computation). We call this a 'performance hit'.

For a large computer M, we will have:

$$f_C \approx 0, f_j \approx 0, \forall j, \; j \neq C, v_1 = 0, v_2 \neq 0 \tag{49}$$

Eq. (46) becomes for a large computer M or a computer sufficiently far away:

$$t_1 \approx t_2 \times \sqrt{\frac{1 - s^2 \times 0^2}{1 - s^2 \times 0^2}} = t_2 \tag{50}$$

A small computer will run slower, but a large computer will not slow-down much.

5.3 Synchronization of Application Clocks

In principle, it's impossible to know exact relative speeds and information-influences of all other computers, because those depend on the results of computations each computer performs and it would take a computing system of a larger capacity than the Universe to know them. However, near large computers, the information-influence of all other computers is very small and therefore in practical terms knowing the performance-hit of applications becomes possible. Assuming that reading of the two same application clocks very near one another can be synchronized to begin with, their readings could then in practical terms be known if the performance-hits are known to a good degree, even when they are separated.

6. Mass and Application Time

6.1 Mass

The time need to process information is:

$$t(i,\Delta i) = i / T_{\text{lossy}} = \left(i / T_{\text{lossless}}\right) \Big/ \sqrt{1-(\Delta i)^2 / i^2} \tag{51}$$

This time can be used to describe a computer. Two computers exposed to the same additional available-information will take different times to react to it. If we denote this measure as mass m, it follows:

$$m \propto t(i,\Delta i) = y \times i \Big/ \sqrt{1-(\Delta i)^2 / i^2} \tag{52}$$

y is a dimensional constant of proportion that absorbs the constant T_{lossless} and produces measurable quantity m. When there is no change in available information:

$$m_0 = y \times i \tag{53}$$

As in Eq. (48), the measure varies with speed in a simple two-computer scenario:

$$m_1 \approx m_0 \Big/ \sqrt{1-v^2 / c^2} \quad . \tag{54}$$

Mass is proportional to the time needed to process the available information.

6.2 Changing Distance

Let us observe a computer C at rest at some distance from a computer M. Any physical volume that C occupies can be split into smaller volumes, where each such smaller volume contains some information from M. The more information of M is present, the more is added to data of C. Thus, when distance to M changes, the change in data is proportional to the available-information of M. The longer C is present at the same location, the more of M's information will add to C's data:

$$d(\Delta i_C) = w \times \frac{i_M}{R^2} \times i_C \times f_M \times dt \tag{55}$$

where $d(\Delta i_C)$ is change in additional data of C; w is the constant of proportion; i_M is the data of M; R is the distance between M and C; f_M is the information influence of M; i_C is the data of C, and $i_C \times f_M$ is the portion of C's information that comes from M; dt is a small enough time interval.

If C moves slowly from very far away to near M, it will see increasingly more information added to its data. We will model a situation of a practical infinite distance between M and C with the speed of zero, meaning there is no change in information processing. When distance between the two declines to R we will model this with some speed v (for the same effect). The change of speed dv will cause change in additional data of C. From Eq. (13), the change in additional information is (information influence f_M is some constant value f, so we substitute $S = s \times f$):

$$d(\Delta i_C) = S \times i_C \times dv \tag{56}$$

After substituting constants ($W = w / S$), from above two equations:

$$dv = -W \times \frac{i_M}{R^2} \times f_M \times dt = -W \times \frac{i_M}{R^2} \times \frac{dt}{1+(i_C / i_M) \times R^2} \quad . \tag{57}$$

where W is a dimensional constant of proportion; i_M is the data of M; i_C is the data of C; R is the distance between C and M. The minus sign is for higher v with lower R.

Multiplying both sides by v, and knowing that $dR = v \times dt$ we have:

$$\int_v^0 v \times dv = -\int_R^\infty W \times \frac{i_M}{R^2} \times \frac{1}{1+(i_C / i_M) \times R^2} dR \tag{58}$$

and

$$v^2 = 2 \times W \times i_M \times \left\{ \frac{1}{R} - \sqrt{\frac{i_C}{i_M}} \times \left[\frac{\pi}{2} - \arctan\left(\sqrt{i_C / i_M} \times R\right) \right] \right\} \quad . \tag{59}$$

As in Eq. (48), and by substituting $G = W / y$ and assuming $i_C \ll i_M$ and $R \gg 0$:

$$v^2 = 2 \times W \times i_M / R \tag{60}$$

$$t_2 \approx t_1 \Big/ \sqrt{1 - 2 \times W \times i_M / R \times c^2} \tag{61}$$

$$t_2 \approx t_1 \Big/ \sqrt{1 - 2 \times G \times m_{M0} / R \times c^2} \tag{62}$$

This is the performance-hit due to its mass m_{M0}.

7. Relativity

The principle of relativity states that laws of physics are the same in un-accelerated frames of reference. Now think about it in different terms: a computer produces the same output if the input is the same. This output would be created slower or faster, depending on the performance-hit, but overall will be almost identical. Note however, that the information loss will add differences to the output of computations, even under conditions of the principle of relativity. This means that relativity (in any known form) must be an approximation only.

Acceleration changes the speed, which changes the degree of information loss. It means that the output of computing will change too. This is the fundamental reason why accelerated systems are so apparently different than inertial ones.

7.1 Speed of Light

An assumption about constancy of speed of light is unnecessary. As shown, the need for local speed limit can be easily derived. Also shown is the necessity for this speed to be the same regardless of the speed of the emitter.

7.2 Time Dilation

The concept of time dilation, as explained in SR/GR theories, is suspect. The performance-hit of computing provides an intuitive, postulate-free and mathematically more generic framework for the phenomena observed.

7.3 Mass Increase

We found that the mass is a simple consequence of computation reality. If a computer needs more time to compute the available information, it has higher mass.

7.4 Maximum Speed

We showed that the speed of light comes from the limited computational throughput. The speed of light is the minimum of all possible maximum speeds. This minimum is in effect only near large masses, or for very small masses.

8. Quantum Mechanics

8.1 Information Loss and QM

Information loss in computing causes its results to be approximate, and have an element of randomness. If computation produces a set of independent output results (such as vector of movement and the speed), then the probability to predict correctly all of them is:

$$p = \prod_{1}^{V} p_i \tag{63}$$

The probability p_i is the probability to predict one of the independent V outputs. It is proportional to the number of facts in the lossy set; Δi^2 that affects any given output result:

$$p_i = (u_i - e_i) / u_i \tag{64}$$

e_i is the part of lossy Δi^2 that affects a given output result V_i and u_i is the part of total processing i^2 used for V_i. It must be:

$$\sum_1^V e_i = (\Delta i)^2 \quad , \quad \sum_1^V u_j = i^2 \tag{65}$$

8.2 A Simple System

In a simple case of two independent output results being vector momentum and speed

$$p_1 = 1 - e_1 / u \quad , \quad p_2 = 1 - e_2 / u \quad ,$$
$$e_1 + e_2 = (\Delta i)^2 \quad , \quad \Delta i < i \quad , \quad u = i^2 / 2 \quad . \tag{66}$$

From this, the maximum probability of predicting both is: If the error in predicted value is inversely proportional to the probability of predicting it, then :

$$p_1 \times p_2 \le \left[1 - (\Delta i)^2 / i^2 \right]^2 \tag{67}$$

With near-maximum information loss (*i.e.* $\Delta i = i - 1$):

$$p_1 \times p_2 \le \left[1 - (i_{\min} - 1)^2 / i_{\min}{}^2 \right]^2 \tag{68}$$

$$\Delta V_1 \times \Delta V_2 \ge (i_{\min})^2 / 4 \tag{69}$$

ΔV_1 and ΔV_2 are estimated errors in predicting the two output values (vector and speed of movement); $i_{\min}$ is the minimum number of facts any computer needs to produce V_1 and V_2, so: Since the dimensional constant of information had been set to exactly 1 and unrelated to any other system of measurement [(in Eq. (1)]

$$\Delta V_1 \times \Delta V_2 \ge \text{constant} \quad . \tag{70}$$

This is the simplified derivation of Heisenberg principle of uncertainty from the computational approach. This basic idea of quantum mechanics is a consequence of lossy computation.

The random component of a movement of a computer (in three-dimensional space) due to information loss allows it to 'triangulate' the vector of other computers; *i.e.*, the direction to other computers. In previous analysis, we focused on scalar nature of information. It turns out that the very nature of information loss provides the means for three-dimensional computing.

If it is the facts that determine the output of computation, then ultimately the output itself has to be quantified in its basic form. Think of it this way: the result of computation, due to limited storage, is essentially an integer, a quantified value. As such, on the basic level, the result must appear in quanta as well.

9. Gravity

Let us have two computers A and B and focus on what happens if B moves back and forth along the line connecting A and B , so that it stays in place over time (see Figure 3).

When closer to A , the information loss of B is higher, and *vice versa*. On average, the loss in computing the direction of movement (from B_1 and B_2 back to B) will result in positions B_3 / B_4 and B'_3 / B'_4, with higher directional loss closer to A (Fig. 4):

$$\Delta L > \Delta L' \tag{71}$$

The result is that, on average, B will move closer to A in a straight line, even with the best effort of B to stay in place. This is not the result of any purposeful computation, only of uncertainty in computing the direction of movement.

The speed of B can be found by equating the information loss to that due to some speed v (under limiting-case assumptions of Sect. 6.2):

$$(\Delta i)^2 / i^2 = 2 \times G \times m_A / R \times c^2 = v^2 / c^2 \tag{72}$$

and

$$v = \sqrt{2 \times G \times m_A / R} \tag{73}$$

B will accelerate towards A and *vice versa.* We have derived Newtonian gravity.

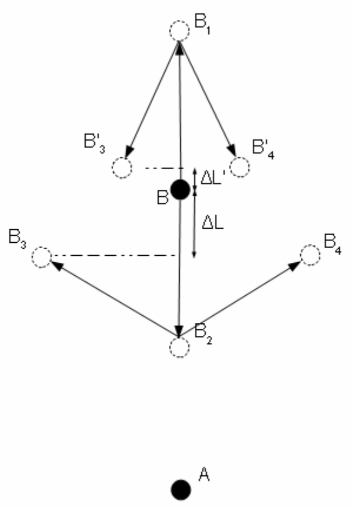

Figure 3.

9.1 Principle of Equivalence

Gravity is shown to result from loss of information processing. This means gravity does not consume any useful information. The direct consequence is that all computation (minus any loss of information) will remain the same. This is the reason why any experiment appears the same whether the laboratory is inertial or in free fall, minus any changing loss of information. The Principle of Equivalence is a consequence of the informational nature of gravity.

9.2 Gravity and Motion

Because gravitational acceleration is due to the directional loss of information, it follows that gravitational acceleration is not necessarily tied to mass *per se*. Remember that movement creates the same information loss, so a localized directional loss can be produced that way alone, *i.e.* a free-fall in empty space can be achieved without massive bodies to claim responsibility for it.

10. Conclusions

We saw that the postulates and results of special/general relativity and other fundamental outcomes can be naturally developed from a simple and intuitive model of reality. The Universe is in essence a distributed computing network, with each node of limited computational throughput and information storage.

It is worth repeating the idea from the Introduction: Without information and the means to use it, the physical reality has no basis for existence. This paper shows a few of the most important (and until now, conceptually divergent) discoveries of physics derived from a simple informational approach, without the need for postulates and principles. The important takeaway is that these postulates and principle are not elementary at all. They are the consequences of the deeper and more profound nature of reality, one that rests on simple and intuitive idea of an informational reality.