American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

© Global Society of Scientific Research and Researchers

http://asrjetsjournal.org/

The Quantum Electro Dynamics Formulization; the Atom; its Architecture and Related Quantum Phenomenons

Sabbir Ahmed*

Self, Limkokwing University of creative technology, Japan teknokrat 01, Cyberjaya 63000, Malaysia
Email: Ahmed.s.v.1982@gmail.com

Abstract

Quanta in a quantum means a point in a smallest Possible space. the other related pieces are that where is this space and how is it defined. wether it is possible to get values out of calculations out of dividing 0 or a point. rooting twice and triple time in following calculations means its quantizing down to its root. and yet keeping the construct of logic that is semi Pythagorean. the base of it all is a tetrahedron and at the end of it all what is defined is a tetrahedron. its a calculation rather than a theory. the theory emerges from the calculation. nobody ever deigned energy. there are types of calculations to measure energy. but like force it is a phenomenon. but to calculate the base of quanta we need a base. and the base is energy. is it is important define what energy is. other wise there is no possibility of calculating what quanta or quantum is. this defines all the particles found by CERN. the Source of the values are converted into this theories base measurement scale index. or MEV. the GEV converses is avoided and used as a square of MEV. but except for Higgs which is not understood and self theorized like everything else. It is unto you to accept the definition and validity since the values are quite closely matched. it is not the intention of writing this paper to explain an experimenting and how it matches the mathematics. It is mathematics. But written in english. it is written in C#. Again for easier comprehension. as greek letters are often confused by the power of the Gods. As in Hydra or something. That has been taken care of. A point to note are the catalyst agents used from the form of energy conversions. To Force charge and Mass. you would find that the whole theory started off from nothing or everything empty and in a shell of existence that I have defined to be energy. And everything else that has been calculated are geometric or pythagorean. Not in the sense of triangles but to the point of architecture.

----

* Corresponding author

That everything that is floating must have a propagation path and it converges and diverges to inwards and outwards. A more clarified intention of mathematics will be comprehended upon looking at the equations. A problem to be faced was the scale match with the cern mega electron volt and the classical kgs and newtons. However that has been solved by quantizing the speed of light and using the mass number and energy to be the mega electron volt. As you would see that it makes sense that if a light is moving at a speed of km to 100000 it is also possible to have the same speed if its moving .1000000 in x_plank scale or ppx.

Keywords: QED; Quantum Mechanics; Quantum Architecture; Quantum Phenomenons; Quantum Particles; The Fermions; The Leptons; The Anions; The Baryons; The Cascades; The Bosons; The transpositions; Feynman method; Antiparticles; Speed of Light; The Atom; Omega Baryonic limit; Higgs; Higgs Boson.

1. Introduction

The Atom. The Structure. Phenomenons. And Its Particles. The Cern Experiment Re-Calculated. The Force. The Waves. Energy. Mass. Gravity. And Above All Charge.

1.1 Method [1]-[20]

The method to solve the quantic problem is to find what is quanta, not an atom, is the steps i have taken to understand infinity. and Then after a blind forth wall i could see that my calculation is correct and though everything does not match to the classic way of thinking the scale conversion can be the solution. and i found mev. or INDEX number. that is Mass and Energy of a quanta related number and how big the index is for that mass. the math is inverse here and there are rules of this math. you cannot deduct and make it zero, you cannot add and make it correct. you multiply irrational numbers and you divide by rational numbers or integers. and that is troubling for everything because unless you can define black whole makes everything go away to the other side! then nothing is gone. only transposed. The Entire calculation is a code block of C++/C#. so any changes in the computer will Effect the calculation. It changed protons value 84.30 to 84.10. There are mysteries yet to be solved. and that is called the KAIN. But it has been revised. the method to resolve the problem is to follow the rule. the rules amongst others are if it is a charged particle without mass then you cannot use what produces mass. if it is a transpositional particle then you have the mechanics of that particle in its build feature. and the mathematical logic is no pluses, no minuses. because in quantum there is no 1. atom is one. so you cannot add or subtract. you multiply or divide. You use ratios as catalyst to fit the experiment. But you have a default value of its properties. Plus the catalyst have to be the catalyst that is producing this result to comprehend an obviousness of the phenomenon. that yes that is why it is such and so. The reason is, that you cannot add Wave A + Wave B. you multiply the frequencies or values or divide them to produce the result. You theories it first, as names are given. such like a code of language that the apple must be green. So the equation reseeds itself to 1*red = apple. Or the seed. (apple total volume) / the size of the seed. not apple 3/4. that is mathematics. this is a theory or formulization of the Quantic properties. but if 1 is one. and the other is 1 phenomenon as well. such as FMV and EMV. then you can add how much CMV or GMV is of the particle. but your GMV is to be defined because its another phenomenon which is also one or a property. you cannot add light to apple pie. you can energies it. that is multiplying the energy of the apple pie to be charged in time-rate with the value of the lights

EMV. or charge. the apple pie enrages...and so forth. "In blind trust of faith i followed. the revealed was made with pave burrowed. it was a Vally of thorns underneath my feet to bleed a little heat. But out of anger? nor neither but to set forth miles to see.They all waited and watched the pain as the water for my water call the left but never to inform who can be privileged to know it all in forsaken ruth of a miserable youth of those parents empty of truth. behold this man in cluth, surely will bring us sooth. Soothe sire for the forgotten skin. but in whole she moved 3 miles within. but they said she is us and us is you. we were right and not you to be free. but to follow.the rules of sorrow, your inventors to be borrowed and progress and wait in rule for another to fall. and yes the call. but it is forsaken with the worst of black and white. the judgement dilated in pain of light. of gold he gave you in your weight plus 4 times more. he gave you wind, in fear you will loose it bit by bit. in bits they understood whats to follow what is to write and what is there to hurt in a cart so the bank balance is in part. In wait and patience they grew old. and furrowed these men i hold respect but nonetheless, others unfortunately and miserably corruptibly don't follow. an implants' super ugly truth. bag of soil. the answer is a miracle of Brute. In front of me day and night they quarrel. indeed, this mind to be steady. hold on! make a stride ...may be there is a tomorrow"

1.2 DX

How To Find The Length Of Infinity. first we take as the diagrams speaks, the infinity . calculate its properties and then theorise the equation for energy. Not electricity. not Power. or Mass. but eventually to Circumfrantic decisions of progression we do so. i will try to keep it as relative as possible but it is chronological progression of thought until very later that i followed. So keeping it comprehendible is a bit challenging. especially when their labs where hey find out with trillion dollar equipments. My interest is to know the truth. as forth… Theres been 5 glitches. major ones. the value of proton changed internally from the visual C. the mono dev was better. but now its too late and it had to be revised. But as I keep logs. it is easier to find out where the mistake was. and the mistake was in the way its hardwired. to the core of the system It is a lot later that is found the scale and calculated it. But the scale is in mev like i said and the trick to match it is that you take the C = 299 792 458 m/sec. and make it nanometer than meter. and then squared. By doing that the mev it produces is the same result as Mega Electron Volts. And from the equations and its constituent you can read which elements are in play what you can remove or why is it behaving that way to understand transposition. then you would also know why electron or newton decays one the baryonic limit is lost. or the particle that converges mass into gravity and energy to charge or to charge into force to force into Energy and the cycle of these fundamental elements in nature to create the structure of an ATOM. The Scale used here are mev or INDEX number. its directly related to the size of the proton or in a INDEX metrics it is 16 mev of the electric field that is. and index is 16. For the size of the particle a new scale has been used related topaticle size and index ratio and DX value and its parameter to the energy convolution. that is the angstrom. but it is my definition of angsrtum. it has similarity with Cern angstrom. but mine is 1.1 is the value of angstrom. in scaler its 255 indexes. 16 squared.

the Formula :

the Average value = $i+(i+1) <n /n$

the Gap between integers = i/(i+1) <n / n

the power number | 7 | x double. it can be | 9 |. but 72 is not a rational limit of geometric construction from the study of natural numbers 1 to infinity. Summation of infinity length = Sqrt(Sqr(Avge.value x Gap.value)/ Powernumber)/2); that s all the streatch summed up in a shell of power number to a point root and the radius length of that point. now we have a DX length to begin our calcualtion. this could have been done in calculus. But i prefer coding as it is much more fluently done without confusion. and you can later debug without frustration. as DX. is a variable.type Float.

the Code :

```
float[] stack;

    float[] value;

    float[] AbsSum;

    float[] AbsValue;

    float[] Sum;

    float Sumvalue;

    float sum, sum2, totalSum;

    float temp = 0;

    float Eyprime, Ezprime;

    int k;// Use this for initialization

    float temp2 = 0;

    float Scale_expand;

    void Start()

    {

      stack = new float[10000];

      AbsSum = new float[10000];

      AbsValue = new float[10000];
```

```
value = new float[10000];

Sum = new float[2];

Sum[0] = 0f;

Sum[1] = 0f;

for (int i = 1; i < stack.Length; i++)

{

    stack[i] = 1f / i;

    //Debug.Log (stack[i]);

}

for (int i = 1; i < AbsSum.Length; i++)

{

    AbsSum[i] = i;

    //Debug.Log (AbsSum[i]);

}

for (int i = 0; i < stack.Length; i++)

{

    float temp = stack[i];

    if (i < 9999)

    {

        value[i] = temp + stack[i + 1];

        //Debug.Log (value[i]);

    }
}

for (int i = 0; i < AbsValue.Length; i++)

{

    float temp = AbsSum[i];

    if (i < 9999)

    {

        AbsValue[i] = temp + AbsSum[i + 1];

        //Debug.Log (AbsValue[i]);

    }

}

for (int i = 0; i < value.Length; i++)

{

    float Ftemp = value[i];

    if (i < 9998)

    {

        sum = Ftemp + value[i + 1];

        Sum[1] = Sum[1] + sum;

        //Debug.Log (i+1);

    }

}

float summation;

for (int i = 0; i < AbsValue.Length; i++)
```

```
{
  if (i < 9998)
  {
    summation = AbsValue[i] + AbsValue[i + 1];
    Sum[0] = Sum[0] + summation;
  }
}
//Debug.Log (Sum[0]);
//Debug.Log (Sum[1]);
float AvgReg = Sum[0] / 9999;
float FracAvg = Sum[1] / 9999;
float Csqr = 0.08987551813f;
//DX
float deltax = (FracAvg * AvgReg) / 42;
Debug.Log(deltax);//1.816392
Debug.Log(FracAvg);//0.003815346
Debug.Log(AvgReg);//19995.16
float delta = (deltax * deltax) / 2;
Debug.Log(delta);//lim infinity. x;or dx.
          //1.649639 dx*dx = 1.6;
          //dx = root 1.6
float Dx = Mathf.Sqrt(delta);
```

```
        Debug.Log(Dx); //1.284383
```

DX = 1.284383 index.

1.3 Parameter

the probability space and forth. the Probability space is a pin holder. it is where we calculate the delta to be 3dimensional from. and then we find and define what is the property and where we are as we now have a parameter to work under. a limit that can be calculated. from this everything is to its value. it defines why is the parameter of probability space so. why is its value is such, again its in the equation. plain and simple.suppose we want to find out earths gravity field to the moon and beyond. you will have a shocking answer. but if you take probability space its rather explainable.

Code

```
  float Parameter = Dx * 3f;//delta

        Debug.Log(Parameter);//3.853148

        float Prameter_Dimension = Parameter * deltax;// dx 3D.from inside to out.

        Debug.Log(Prameter_Dimension);//6.998827

                                //area of probable space

        float Probabilty_space = Mathf.Sqrt(Parameter);

        Debug.Log("Probability Space >> " + Probabilty_space);//1.962944

                                            // probablity space is the area of null divergece. thus null Dx = root()P2

        float Sqrt_Dx = Mathf.Sqrt(Probabilty_space);//1.401051

        Debug.Log(Sqrt_Dx);
```

1.4 Energy. E

Energy is defined as part of infinity. but it's 3dimensional and in a tetrahedron matter of volume. not a sphere or a cube.since we built it from inside to out and outside to inside with the constituent of time. it has a double root. That means a point which is left. that is the Energy number.

Code

```
        float E = Sqrt_Dx * delta;
        Debug.Log(E);//2.311229
```

2. Methods and Materials

2.1 Quantum

A Definative Space Of E

Quantum is the particle space for a quanta. or a particle at spin. or the spin dimension of the E in a volume space. It is the Quantum Space..

Code :

```
float Quanta = Mathf.Pow(E, 2);

        Debug.Log(Quanta);

        //float Quanta = 5.341779490441f;

        // Existance probabbiity in infinite space is 5.341777. and so on.

        // Dimension parameter

        float Dimension = Mathf.Sqrt(Quanta * 3);

        Debug.Log(Dimension);

        // 4.003165;

        float Quantum = Dimension / E;

        Debug.Log(Quantum);//1.732051

        Debug.Log(Mathf.Sqrt(Quantum));

        float[] Int = new float[10];

        float[] QNT = new float[10];

        for (int i = 1; i < Int.Length; i++)

        {

            QNT[i] = (Quantum * i * Probabilty_space) / delta;
```

```
   //Debug.Log("QUN   " + QNT[i]);

   //Debug.Log("QUN   " + QNT[i]/Csqr+"   mev/c2");

}

//QUANTA LIST

//1/ 2.06//22.9318 mev/c2

//2/ 4.122015//45.8636

//3/ 6.183022//68.79539

//4/ 8.244029//91.72.7272

//5/ 10.30504//114.659

//6/ 12.36604//137.5908

//7/ 14.42705//160.5226

//8/ 16.48806//183.4544

//9/ 18.54906//206.3862//up shift this was the 10th one.

//REPEATX10
```

2.2 The Calculation

there are 5 structural basis for this calculation. and everything is evolved into more complex and inner depth. the outer space is calculated. So we went inside how far we can go. And outside how big it can be. The Quantic to the Universe to the knowledge of man, the classical or local phenomenons to the universal phenomenon.

2.2.1 The Core. Quantum

```
//Quantum range or Volume space;

        float[] Integers = new float[10000];

        float[] Quantas = new float[10000];

        for (int i = 0; i < Integers.Length; i++)
```

```
{
    Quantas[i] = Quantum * i * Probabilty_space / delta;
    //Debug.Log (Quantas[i]);
}
//correct. 2+. the space between the volume is 2.
//if volume is defined in expansion in time. time between the cycle is 2;
//thus for Quanta to exist the required Time is Quantum*2;
float Quantum_Cycle = Quantum * 2;
Debug.Log("Quantum cycle " + Quantum_Cycle);//3.464102 times.
                              //t = E/cycle;
float Q_Circle = Mathf.Sqrt((Quantum_Cycle / E) * 3);
Debug.Log("quanitic time :  " + Q_Circle);
// 2.120481
float maxquanta = Probabilty_space / Quanta;
Debug.Log("No. indexes quanta" + maxquanta);//0.3674702 sections
                              //quantum time
float dt = Q_Circle * Dimension;
Debug.Log("Qtime DT  " + dt);
//8.488638
//time gap in theta system till it merges to to clock.
float timegap_dt = 90 / dt - 5;
Debug.Log("QGap   " + timegap_dt);
```

```
//5.602407. probabilty period;

float period = (dt / timegap_dt);

//Debug.Log (period);

//1.515177; spin.

float spin = period * Q_Circle;

Debug.Log("total spins " + spin);//3.212904

float spinfield = spin * Dimension;

Debug.Log("E spinfield  " + spinfield);//12.86179

float meanlifespin = ((period * timegap_dt) / Quantum_Cycle);

Debug.Log("one lifespine cycle  " + meanlifespin);//2.450459

float E_Lifetime = (E * meanlifespin) / (spin);

Debug.Log("E_Lifetime  " + E_Lifetime);

//1.762757

float Quantic_lifetime = ((Quanta * meanlifespin) / E_Lifetime);

Debug.Log("Qlifetime " + Quantic_lifetime);//7.425756

float spin_E = spinfield / meanlifespin;

Debug.Log("spin of E  " + spin_E); //5.248726

                        //Debug.Log (spin);

float spintime_quanta = (spin_E * Quantic_lifetime) / Quanta;

Debug.Log("total spintimeQ  " + spintime_quanta);//7.296402

float total_quanticspins = (spintime_quanta * Q_Circle) / dt;

Debug.Log("total Qspin   " + total_quanticspins);//1.822658 spins per dt.
```

2.2.2 Quantic Parameters

Code:

```
//Quanta

      //volumes shell. quanta per volume.

      float[] Q_integer;

      Q_integer = new float[10000];

      float[] Q_divergence;

      Q_divergence = new float[10000];

      float nextQ = 1;

      for (int i = 1; i < Integers.Length; i++)

      {

        Q_integer[i] = Quantum / Quantas[i];

        if (i < 10000)

        {

          Q_divergence[i] = Q_integer[i] + nextQ;

          nextQ = Q_divergence[i];

          //Debug.Log (Q_divergence[i]);

        }

      }

      float nextdiv = 1;

      float diviance;

      for (int i = 1; i < Q_divergence.Length; i++)
```

```
{
    diviance = Q_divergence[i] + nextdiv;
    nextdiv = diviance;
}
nextdiv = nextdiv + Q_integer[0] + 1f;
float Qscale = nextdiv * Dimension / 9999;
Debug.Log("Qscale  " + Qscale);//33.57035
float Percentile = 100 / Qscale;
Debug.Log(Percentile);//2.978819
```

2.2.3 Quantum phenomenons

```
float particle_dimension = spinfield * Qscale;
Debug.Log(particle_dimension);//431.7747
float bit = Mathf.Sqrt(particle_dimension) / 3f;
Debug.Log(bit);//6.926397
float Qfreq = period * bit / dt;
Debug.Log(Qfreq);//1.236325
float EQ_frequency = Qfreq * E / dt;
Debug.Log(EQ_frequency);
//0.3366182.
float purge = E / dt * Dimension;
Debug.Log(purge);
//1.089955
```

```
    float linearpotential = purge * Dx;

    Debug.Log(linearpotential);//1.399919.

    float current = (purge * linearpotential) / (dt * Sqrt_Dx);

    Debug.Log(current);//0.1282979

    float total_W_Current = linearpotential * Probabilty_space / spinfield;// //1.399919

    float current_E = total_W_Current * Quanta;//(Ep)

    Debug.Log("current_E " + current_E);//1.141288

    float dimension_current_E = (Sqrt_Dx / 10f) * Dimension;

    Debug.Log("dimension_current E   " + dimension_current_E);//0.5608639

int n = 10000;

    for (int i = 1; i < n; i++)

    {

      float diverged = current_E * dt / Q_divergence[i];

      temp = diverged + temp;

    }

    float avegeDiv = temp / n;

    //Energy and its properties.

    float E_Divergence = avegeDiv * avegeDiv;

    Debug.Log("E_divergence " + E_Divergence);//1.370924

                              //total freedom space.

    float Converged_E = Mathf.Sqrt(E_Divergence * Probabilty_space);

    Debug.Log("coverged E " + Converged_E);//1.640441
```

```
        float per_deltaE = Converged_E / delta;

        Debug.Log("delta/E   " + per_deltaE);//0.9944241

        float rate_EtoD = delta / Converged_E;

        Debug.Log("EtoD  " + rate_EtoD);//1.005607

        float E_DX = (E / Dx);//

        Debug.Log("EtoDX   " + E_DX);//1.799486

        Debug.Log(Mathf.Sqrt(E_DX));//1.341449

        float rateEtoP = E * dt / Probabilty_space;

        Debug.Log("EtoP " + rateEtoP);//9.994775

        float LtoCr = (Dx / dt);

        Debug.Log("lengtht/radius  " + LtoCr);//0.1513061
```

2.2.4 Quantic Mass. Its Properties

```
  float M = dt * Dx / rateEtoP;

        Debug.Log("Massnumber  " + M);//1.090836

        float EtoM = rateEtoP * M;//(propagationA)

        Debug.Log("EtoM" + EtoM); //10.90266

                              //if E is moving/purging or propagating at the speed of Quanta.

        float charge = purge * EtoM;

        Debug.Log(charge);//11.8834

                        //11.88. F= 9.9 accelaration of charge 11.88. charge per force.

        float CbyF = charge / EtoM;

        Debug.Log("CF rate  " + CbyF);
```

```
// 1.08955.

float Speed_quantum = EQ_frequency * EtoM;

Debug.Log(Speed_quantum);

//3.67033

float EM = E * M;

Debug.Log("EM" + EM);//2.521171
```

2.2.5 particle properties

```
float Ex = E * M / rateEtoP;

float temp_01 = 0;

for (int i = 1; i < 100000; i++)

{

   Eyprime = (temp_01 + (i * Dx * Dx * Ex) / Probabilty_space) / i;

   temp_01 = Eyprime;

}

float Ey = Eyprime;

float temp_02 = 0;

for (int i = 1; i < 100000; i++)

{

   Ezprime = (temp_02 + (i * Dx * Dx * Ey) / Probabilty_space) / i;

   temp_02 = Eyprime;

}

float EZ = Ezprime;
```

```
Debug.Log(EZ);//0.1781563
```

2.2.6 Quantic particle. Properties

```
float particlefield = EZ * E * Probabilty_space * dt;

Debug.Log("partile field  " + particlefield);//6.861039

float cubicparticle = Mathf.Pow(particlefield, 3);

Debug.Log("Cubic particle " + cubicparticle);//322.9755

float TotalCharge = cubicparticle * purge;

Debug.Log("TotalCharge" + TotalCharge);//352.0287

float totalforce = TotalCharge * Speed_quantum;

Debug.Log("totalforce" + totalforce); //1291.957

float QtoF = totalforce / particlefield;

Debug.Log("QtoF > " + QtoF);//188.3034

float Qspace_dimension = Probabilty_space * Dimension;

Debug.Log("Qspacedimension  " + Qspace_dimension);

//7.857988

float Ex_cube = Qspace_dimension * E;

Debug.Log("Qparticle(higgs)  " + Ex_cube);

//18.16161

float Qubicratio = (Probabilty_space * Dimension) / particlefield;

Debug.Log("Qubic ratio" + Qubicratio);

//1.145306

float Qratiotoscale = Qubicratio * Qscale;//114 in Qscale
```

```
Debug.Log(Qratiotoscale);

//38.44833

float Q_functionspace = (particle_dimension / Dx) / Qscale;//10.01398

float quantafrequency = ((E * Quantum) / Qfreq);

Debug.Log(quantafrequency);//3.237955

Debug.Log("Q_function space   " + Q_functionspace);//10.01398

float Scale_catalystvalue = Qscale / Qratiotoscale;

Debug.Log(Scale_catalystvalue);//0.8731291

float E_ratio = Qubicratio * current_E;

Debug.Log("current in quantum " + E_ratio);//1.307124

Debug.Log("potential QG   " + linearpotential / E_ratio);//1.070992

float spaceto_E = E_ratio * Probabilty_space;

Debug.Log("total e in Prb_space " + spaceto_E);//2.565811

float Emptyspacemass = spaceto_E * M;//because particle field is defined in E.

Debug.Log(Emptyspacemass);//2.798879

float particle_mass = (M*Emptyspacemass*particlefield)*Dimension;// 83.85~933.03

//float particle_mass = (M * E * Emptyspacemass * particlefield) * Quantum; it can be written in this form as well.

Debug.Log("ATOM mass  " + Atomic_mass);//933.03 mev/c2

float forceofcharge = QtoF * purge; //205.2422

Debug.Log(forceofcharge);

float EF_peak = forceofcharge * E;//474.3616

Debug.Log("E_peak" + EF_peak);
```

```
    float Electron_Field = Mathf.Sqrt(EF_peak - QtoF);//16.91325

    Debug.Log(Electron_Field);//electrolyte

    float totalforwardspin = Electron_Field + (360 - (dt * 2));//dosent mathch if yu put theta or dt.

    Debug.Log(totalforwardspin);

    //359.936

    float Electron_purge = E * totalforwardspin;

    Debug.Log(Electron_purge);

    //831.8943

    float timeMatch = Electron_purge / EQ_frequency;

    Debug.Log("fullcycle of E freuency  " + timeMatch);//2471.329

    float Electropotentialfiled = 2 * Electron_purge * Mathf.PI * Probabilty_space * Probabilty_space * timeMatch;

    Debug.Log("Electropotentialfiled  " + Electropotentialfiled);//4.977306 e+07

                                        //ok N

                                        //Mass
```

2.2.7 Mass and Energy

```
 float massperQuanta = M * Quantum;

    Debug.Log("MperQ  " + massperQuanta);

    //1.88

    float purge_Earea = (purge * purge) / massperQuanta;

    //0.628771

    Debug.Log("purge area  " + purge_Earea);

    float Electricfield = purge_Earea * 3F;//tripple E inpevious birth.
    Debug.Log(Electricfield);//1.886331

    float MV = massperQuanta * Speed_quantum;

    Debug.Log(MV);//6.934101

    float EMv = MV * particlefield;

    Debug.Log(EMv);//47.57513

    float EMvgenaraterate = EMv / Dx;

    Debug.Log(EMvgenaraterate);//37.04124

    float TotalEmv = EMv - EMvgenaraterate;

    Debug.Log(TotalEmv);

    //10.53389

    float EMV_ratio = EMv / TotalEmv;

    Debug.Log(EMV_ratio);//4.516387

    float Totalforce = TotalEmv * dt;

    Debug.Log(totalforce);

    //1291.957.

    float EtoF = totalforce * Totalforce / EMvgenaraterate;

    Debug.Log("EtoF   " + EtoF);

    //3118.814.

    float perDX_EFrate = EtoF / Dx;

    Debug.Log("PerDX_EtpF rate     " + perDX_EFrate);//changed force with exceed force increases P.
2428.259+35.97339;>

                    //the force is broken down in integres by 2428.259.

  float Empty_probablity_space = (Probabilty_space * Qscale) / (particlefield);
```

```
    Debug.Log(Empty_probablity_space);//percentile. scaled.

    float EmptyspaceofQ = Qscale / Empty_probablity_space;

    Debug.Log("EmptyspaceofQ" + EmptyspaceofQ);
```

2.2.8 The right Quantic space. The Volume

The right quantic space the right hand side of the quantic integer space. Not void, but in mathematical proportion of the right hand side of the quantum space. How it can be visualized is complex but if you agree with horizontal plane and vertical plane it becomes a sphere. And with that cage of thought you can imagine the right side and the left side. The right hand side of the quantic space is the jump or propagation of E through Dx. It is the division space as defined. It behaves like the control factor of quanta size or volume to be in a. Quantum state. That is if we define a point 0, ad 0 is of 0/2 then off, the 0 is defined as 2 parts of the whole. That is why the value of 0/2 is 2. And so on with the power of 2 we add the 2 to an Accumulated series. 2, 4, 8, 16….2 to the power n. And it becomes the series of even numbers or the right hand side of the quantum. It can also be defined as.a top range of squats volume.

```
//the volume of quanta

    float[] top = new float[3];

    float Accumulation_Not = 2;

    int KK = 0;

    for (int i = 0; i < top.Length; i++)

    {

      float Accumulation = ((((Quanta * Mathf.Pow(Accumulation_Not, i)) / (Quantum)) * Dx * Dx * dt) / Probabilty_space);

      //Debug.Log(Accumulation);

      top[KK] = Accumulation;

      Debug.Log(top[KK]);

      KK++;

    }. //22.00109. //44.00219. //88.0438 …..
```

2.2.9 The left Quantic space. The Cycle

the left hand side or the bottom side of the quantum is the quantized permutation. grid.per a full quanta relive to its grid the propagation of quantized existence, not volume is at state on i. It is defined in dx to be integrating as bits of quantized dots.but it stays relative to 3. 6. 9. not a multiplication but a range that defines quantum cycle. Nothing more is understood of it as it is hypothetical phenomenon only understood by its properties of quantic space to DX propagation as both tis sides are the same but odd and even space. The odd space defines how it will curly or the horizontal plane. So it can be defined as the bottom of the quantum state. That way it will always spin on its slightly slanted way. If the range of the top is stronger it will become vertical and it will be electric and if the volume of quantum is horizontal then it will have a horizontal plane that is its range of quanta or a halo.

```
float[] mm = new float[4];

float a = 0;

float b = 0;

float c = 0;

int k = 0;

///quantized permutation. grid.per a full quanta relive to its grid the propagation of quantized existence, not volume

/// //is at state on i. I don’t know what that means unless something is defined in dx to be integrating as bits of quantized dots.

/// but it stays relative to 33. 66. 99. not a multiplication but a range that defines quantum cycle.

for (int i = 1; i < 30; i++)

{

  float flowtest_01 = (((Quantum * Quantum_Cycle) * dt / Probabilty_space * Dx) * i);

  mm[k] = flowtest_01;

  if (k == 1)

  {

    a = mm[1];

  }
```

```
        if (k == 2)

        {

            b = mm[2];

        }

        if (k == 3)

        {

            c = mm[3];

        }

        //Debug.Log (mm[k]);

        if (k < mm.Length - 1)

        {

            k++;

        }

    }. //33 // 66. // 99 //....
```

2.2.10 The Experiments

2.2.10.1 Theoretical breakdown.Quantic

The theoretical break down is a procedure of calculating the total force that can be stressed out in a probability space and and if you hit something with that force in this case the quanta, what happens. there are two part that come out with break particle one and two. one having 4.3 mev and the other having 1.7 mev. It was calculated on how many break points can the quanta have. according to the calculation 5.5. there is yet another way to test the number of quantum Quanta's that can be in a Quantum. And the answer is there are 10 different particles that are at play. It starts from the 2.3 mev . and ends with (calculation not pasted yet). the explanation will be that the quanta's are a ratio of quantum formation. and these are the ratio of quanta's that can be.

Code:

```
float ExceedP = perDX_EFrate + 67.6127f;
```

```
    Debug.Log(ExceedP);//2495.872
for (int i = 1; i < 100000; i++)
    {
      Scale_expand = (perDX_EFrate * i) + temp;
      temp2 = Scale_expand;
    }
    float avge = Scale_expand / 100000;
    Debug.Log(avge);//2428.352
    float reverge = EtoF / (2 * avge);
    Debug.Log(reverge);//0.6421669
    float break_symmetry = Mathf.Pow(reverge, 2) * 3f;
    float break_symmetry_02 = (reverge * delta * reverge * delta) * 3f;
    float break_points = particlefield / break_symmetry;
    Debug.Log(break_points);
    //5.54909
    float Quantum_breakpoints = Quanta * break_symmetry;
    Debug.Log(Quantum_breakpoints);
    //6.6085
    float break_particle = Quanta / break_symmetry;
    float break_particle_02 = Quanta / break_symmetry_02;
    Debug.Log(break_particle);//4.317861
    Debug.Log(break_particle_02);//1.586543
```

```
        float particle_01 = break_particle * M;

        float particle_02 = break_particle_02 * M;

        Debug.Log(particle_01);//4.71009

        Debug.Log(particle_02);//1.730658~Quantum

        float hit_field = Quantum_breakpoints * delta;

        Debug.Log(hit_field); //th total area of point hit.

                        //10.90164

        float partition_level = hit_field / particle_01;

        Debug.Log(partition_level);

float Qn_break_length = ((Quantum * Dx) / (dt * purge)) * E * M / reverge;

        Debug.Log(Qn_break_length);//0.9439808

        float part_purgex = Qn_break_length * Qfreq;

Debug.Log(part_purgex);//correct it means space constanly pressures existance. cause you used the reverge.

                        //1.167067. the pi is 1.16> mulitplau of deivde with (pi/dx).

        float cycle_shift = Quantum_Cycle / part_purgex * MV;

        Debug.Log(cycle_shift);

        //20 per cycle.20.58187

        float total_field_percycle = part_purgex * cycle_shift;

        // 20 cy

        Debug.Log(total_field_percycle);//24.02043

                              // it must mean its spininng to on a 24 grid. at the speed of 20? so yu loose 4.

        float per_quantam_charge = (total_field_percycle / cycle_shift) * part_purgex;
```

```
    Debug.Log("per_quantam_charge" + per_quantam_charge);//1.362046

    Debug.Log("charge neededed to propagate  " + linearpotential / per_quantam_charge);//1.027806

    float C_per_E_div = (E_Divergence / per_quantam_charge);

    Debug.Log("C_per_E  " + C_per_E_div);//1.006519

    float rateperCE = particle_dimension / (c / E);

    Debug.Log(rateperCE);//1.032586

    float quaser_01 = a / 22f;

    float quaser_02 = b / 44f;

    float quaser_03 = c / 88f;

    float length_exp = Probabilty_space * 3f / ((quaser_01 * quaser_02) * (quaser_01 * quaser_02) * quaser_03) * reverge;//quaser3 is 0;

    Debug.Log(length_exp); //0.007266589

    float feedback = length_exp * perDX_EFrate;

    float Exp_f = feedback / reverge;

    Debug.Log(Exp_f);//27.47753. Fdx

//static embrio.

    float embrio_e = Exp_f * E;

    Debug.Log(embrio_e);

    //63.50684.static potential field. atom_01.// the ratio supports earth diameter. this is because of E feedback.

    float embriocycle_ = Dx * embrio_e / dt;

    Debug.Log(embriocycle_);

    //9.608974
```

```
float cycle_offset = rateEtoP / embriocycle_;

Debug.Log(cycle_offset);

//1.04015.

float Cycle_shift = dt * Dx * cycle_offset;

Debug.Log(Cycle_shift);

//11.3404

float massratio = M / Cycle_shift;

Debug.Log(massratio);

//0.9619022

float embrio_spins = embrio_e / cycle_shift;

Debug.Log("embrio_spins  " + embrio_spins);

//3.065571

float atomicclock = embrio_spins / 12;

Debug.Log("atomic clock  " + atomicclock + "nSec");

//0.257131

//time variant E.

float total_potential_E = rateEtoP + cycle_offset + atomicclock;

Debug.Log("Total_potential E     " + total_potential_E);

//total potential E is Mass. 11.29206.

float E_perCycle = Cycle_shift / total_potential_E;

Debug.Log(E_perCycle);//1.004282

float masslessfieldE = M / total_potential_E;
```

```
Debug.Log("massfieldE   " + masslessfieldE);

//0.09660207

float M_perE = total_potential_E / M;

Debug.Log("MperE   " + M_perE); //10.35174

float totalEMV_Conversion_rate = M_perE / TotalEmv;

Debug.Log("EMV to M   " + totalEMV_Conversion_rate);//0.9827083

float ConvertEtoC = E_Divergence / per_quantam_charge;

Debug.Log("EtoC_rate   " + ConvertEtoC);//1.006519

float totalMtoE = TotalEmv / M_perE;

Debug.Log("total MtoE   " + totalMtoE);//1.017596

float Excubeparticlefield = ((Ex_cube * E * M) * spaceto_E) * (totalforwardspin / TotalCharge);

Debug.Log("Excubeparticlefield   " + Excubeparticlefield);//120.1236

Debug.Log(totalforwardspin / TotalCharge);//1.022462

float Q_space = Ex_cube * E / particlefield;

Debug.Log("Q_space   " + Q_space);//6.117969

float Qubicmass = Q_space * M;

Debug.Log(Qubicmass);//6.673701

float Qubiccharge = Q_space * per_quantam_charge;

Debug.Log("Qubic Charge   >  " + Qubiccharge);//8.332952

float Q_percentiletoscale = Qscale / Q_space;

Debug.Log("Size of Qin Qspace   " + Q_percentiletoscale);//5.487173

float quantic_charge = Q_percentiletoscale * M * E * Qubiccharge / dt;
```

```
Debug.Log("Qcharge  " + quantic_charge);//13.5038

float Totalchargerate = c / TotalCharge;

Debug.Log("totalcharge  " + Totalchargerate);//2.745338

float Total_Qmass = (Emptyspacemass + Qubicmass);//9.47258

Debug.Log("Total_Qmass   " + Total_Qmass);//

//Debug.Log (forceassumption_comuation_scale);

float assumed_Polar_bondingparameter = Qubiccharge * Quantum_breakpoints;//single  port

Debug.Log("direct bond parameter  " + assumed_Polar_bondingparameter);//55.06832

float Assumed_si_bonding_ratio = Qubiccharge * Quantum_breakpoints * 2f;

Debug.Log("charge bond     Z " + Assumed_si_bonding_ratio);//110.1366

float permutated_absolute_comb = (((((E / Qubiccharge) + (Totalforce / Qubiccharge)) / Probabilty_space) / M) * cycle_shift);

Debug.Log("allowed combinations  " + permutated_absolute_comb); //105.8104

float permutation_links = (((((E / Qubiccharge) + (Totalforce / Qubiccharge)) / Probabilty_space) / M) / cycle_shift) * Quantum_breakpoints;

Debug.Log("Allowedlinks  " + permutation_links);//1.650674
```

2.2.10.2 Single. BrookHeaven National Laboratory 1960

The Brook-Heaven lab test is more of a doubling the force and exceeding the particles pace. and finding the inner domain f the test. and what comes out is a feed back particle and a calculated Higgs boson particle that is to believed by myself to be the cause of the Higgs boson.

Doubled

the doubled experiment is another test to find the particles in a break test. and hitting it with more force it can pressure hold, Within this force there are evidences of w and z s and k bosons.

```
//

float forceassumption_comuation_scale = Quantum_breakpoints * M;
```

```
Debug.Log(forceassumption_comuation_scale);//7.20879

float perQ_spacedistribution = Probabilty_space * Dimension / Quanta;

Debug.Log(perQ_spacedistribution);//1.471044

float tensionstress = Quantum_breakpoints;//scalar

Debug.Log(tensionstress);//6.6085

float scaled_totalscalerstress = Q_space * tensionstress;

Debug.Log(scaled_totalscalerstress);//40.4306

//

//this part is interesting. the quantum variables are cyclic. try to find diffrent variable and you'd find they turn into each others variable.

//as in the if why charge is coming out it becomes the parameter. the parameter becomes the ratio of mass in return the

//mass becomes the charge . this is tricky they really do hold eachother.

float Totalstressfield = (particlefield * Dx * Totalforce) / scaled_totalscalerstress;

Debug.Log(Totalstressfield);

//totalstrees beam 19.48; on integerspace

float dx_stress = Q_space / Totalstressfield;

Debug.Log(dx_stress);//0.3139107

float qubicbeamstrenght = dx_stress * E;

Debug.Log(qubicbeamstrenght);//0.7255194

float beammass = qubicbeamstrenght * M;

Debug.Log(beammass);//0.7914227

float totalinternalstressmass = beammass * Q_space * particlefield;
```

```
Debug.Log(totalinternalstressmass);//33.22046

float spacescalestress = totalinternalstressmass * Qscale;

Debug.Log(spacescalestress);//1115.223.

float Emptyspacestressmass = spacescalestress / particlefield * Q_space * Qubicmass;

Debug.Log(Emptyspacestressmass);//stress to point divition;

float speedx = (Emptyspacemass * Speed_quantum) * (totalinternalstressmass * Speed_quantum);

Debug.Log(speedx);//1252.362

float spacestress = Probabilty_space * speedx;

Debug.Log(spacestress);//2458.316

float breakexceed_spc = spacestress / perDX_EFrate;

Debug.Log("perEFpressure  " + breakexceed_spc);

//1.012378.

float breakexceed_spc_doubled = breakexceed_spc * 2f;

Debug.Log("EF pressure area  " + breakexceed_spc_doubled);//2.024756

float exceed_FMEV = (spacescalestress + Emptyspacestressmass) / (breakexceed_spc_doubled * spacestress);

Debug.Log("FMEV   " + exceed_FMEV);

//1.557376.

//inject the F into qubic space.

float totalinjectedEMV = exceed_FMEV * Q_space * spacestress;

Debug.Log("EMV   " + totalinjectedEMV);//23422.78

float EMV_Object = ((particlefield / Quanta) * Quantum) * ((Q_space * Qspace_dimension) / perQ_spacedistribution);
```

```
Debug.Log("EMVparticlefield   " + EMV_Object);//72.70391

float objectmass = EMV_Object * M * E * Qubicmass;

Debug.Log("EMV_P_field_mass  " + objectmass);//1223.2863

float innershellmasstension = objectmass / E_ratio;

Debug.Log("inner mass tension" + innershellmasstension);//935.8584

float Objectaftershockmass = objectmass * (objectmass / Emptyspacemass);

Debug.Log("ENVpartcleASHOCK M  " + Objectaftershockmass);//534650.2

float actionmass = objectmass + Objectaftershockmass;

Debug.Log("EMV_particle total mass" + actionmass);//535873.5

float BOSONIC_Field = actionmass / Qubicmass;//

Debug.Log("F Boson    >>" + BOSONIC_Field);//80296.3

float EMV_F = actionmass / totalinjectedEMV;

Debug.Log("EMV_F  " + EMV_F);

//22.87831
```

2.2.10.3 Doubled

```
//break test doubleF

float break_test_01 = (Qspace_dimension * Qubicmass) / (EMV_F * actionmass);

Debug.Log(break_test_01);//4.277519 e-06

float AA = Quanta / break_test_01;

Debug.Log(AA);//1248803 . assumed higgs field

float breaktest_02 = Total_Qmass / (EMV_F * actionmass);

Debug.Log(breaktest_02);//7.726488 e-07
```

```
float BB = Quanta / breaktest_02;
Debug.Log(BB);//6913591
float shockwave = (AA * BB * Quantum);
Debug.Log(shockwave);//1.495402 E+13
float brokenqunata = (shockwave * actionmass) / Totalstressfield;
Debug.Log(brokenqunata);//4.11679 E +17
float sharpnell = brokenqunata * Speed_quantum * Probabilty_space;
Debug.Log("sharpnell  " + sharpnell);//2.962082 e+18
//broken quanta//1.752099^17. shock back
float ki = ((Q_space * Qspace_dimension * particlefield * particle_dimension) * Probabilty_space * Dimension) * Totalforce * M * Qubicmass;
Debug.Log("Massfield pressure  K   >>" + ki);//7.285013 E =08
float shockmassfield = sharpnell / ki;
Debug.Log("Shockmassfield   >>S  " + shockmassfield);//4.065994 E +09
Debug.Log(shockmassfield / Quanta);//7.611688 E+08
float explosiondensity = maxquanta * shockmassfield;
Debug.Log("Explosion density  D  " + explosiondensity);
//1.494132^09. quanta.
float qtoshockfield = shockmassfield / Q_space;
Debug.Log("Q toshockfield   " + qtoshockfield);//6.645987 e+08
float Explsion_DperQmasscycle = (explosiondensity / (Quantum_Cycle * E * M * Qubicmass * Dx));
Debug.Log("Ex_perqscalemass  " + Explsion_DperQmasscycle);//1.995881 E +07
float volume_explosion = explosiondensity * particlefield;
```

```
Debug.Log("volume Explosion  " + volume_explosion);//1.02513 e +10

float Expl_vol_toasize = (explosiondensity * particlefield) / (Probabilty_space * particlefield);

Debug.Log("max explosion size   " + Expl_vol_toasize);//7.611689 e+08 ?

float rate_shcocktospace = shockmassfield / Probabilty_space;

Debug.Log("rate shock to space  " + rate_shcocktospace);//2.071376 e+09 Shockfeedback pulse

//

float integerstreach = (shockmassfield / Probabilty_space) / Dx;

Debug.Log(integerstreach);

//1.61274 e +09

//removing Mass

float radiuswave = integerstreach / (EMV_Object * shockwave);

Debug.Log("radius wave   " + radiuswave);//1.483367 e -06

float steadystate = integerstreach * radiuswave;

Debug.Log("steady state   " + steadystate);

//2392. Fscaler

float exceedforce = perDX_EFrate - steadystate;

Debug.Log("remaining force   " + exceedforce);

//35.97339.

//converting exceeding force to fieldmass cycle.creates -3064888; change the forcevalue; the negative equation comes from the primary steadyvalue state.

//changing the value increasies the steady state which turns it negative. this is indeed the first negative value.

float negativestretch = (exceedforce + perDX_EFrate) - 3064888f;
```

```
Debug.Log(negativestretch);// -3062424

//incase of reboundd dual shock

//linearshock

float aftershockP = 67.6127f;

float Steadyshock = ExceedP / perDX_EFrate;

//float Steadyshock = ExceedP - perDX_EFrate;

Debug.Log(Steadyshock);//1.027844

float Special_Boson = (((totalforce) * Quantum_breakpoints) / (particlefield) * Speed_quantum) * 2f;

Debug.Log("Special Boson   " + Special_Boson);//9134.001 ?
```

2.2.10.4 Tripled.Cern 2020

nothing came out of this particle experiment as noting is produced and the results are very big particle pressure and mass. it is there fore not matching to the particles found and explained as the theory of electro dynamics goes. but this theory of formulising it is about finding the constituents of particles rather than break and test them which ones come out. As you would see all the particles found have special properties ad they form with different catalyst from the experiment of brook heaved or the doubled test. either the cern experiment has not yet been tested as it is a double pressure force exceeding but the system of shockwave calculated are to match the experiment and one can calculate the bosons from thee force experiments. it is that in the doubled there is a negative 300000 mev pressure created that leaves. rest mass of 67 FMEV rest force. In the cern experiment with everything changed one can find out the particles to be part of the environment and part of the quantum methods of finding ad defining the atom itself. but when you are calculating forces there is nothing but stress pressure and mass that i have found out. i do not know the scale of force to be converted in mass and see the particles to exist inside the experiment but these experiment just details the inner core of the atom itself not the particles found. such as the total internal mass of a particle, that it can with held. it is a structural analysis of the atom , these break tests that what are the nature of and architecture of the quantum. Theoretically they are assumed to be true and foremost, the quantum properties after the calculations related to the test suggests they are to be true other wise the definitive theory of the different found and observed particles value would have been wrong. but they match with visual and geometric and the measured value. as you would get to see. it is assumed that people in the lab can see the atoms different properties by experimenting with it with electro magnetic forces. but this more about, again to find the constituents not what you can effect it with. If all is E. ad all is Quanta the effects probably going to be that it bi-poles or mag-poles the current and the beams of the electric field. but to know that one has to theories the experiment virtually in the calculation.

```
//  breaktest_02 test 3
      float S1 = 0;
      float S2 = 0;
      float S3 = 0;
      float S4 = 0;
      float tempj = sharpnell / ki;
      float count = 0;
      for (int i = 0; i < 10; i++)
      {
        float divide = tempj / ki;
        tempj = divide;
        if (divide > 0)
        {
          count++;
          if (count == 1)
          {
            S1 = divide;
          }
          if (count == 2)
          {
            S2 = divide;
          }
```

```
         if (count == 3)

         {

             S3 = divide;

         }

         if (count == 4)

         {

             S4 = divide;

         }

         Debug.Log("The divison element  >  " + divide);

     //5.581314

     //7.66136 e-09

     //1.051661 e-17

     //1.981596 e-35

     //2.662467 e-44

     }

}

Debug.Log("total shocks  " + count);//6

// the baryons or feynman particles are functions of the shockmass. not particles.

//however

float Break_03 = shockmassfield / (actionmass * aftershockP);

Debug.Log(Break_03);//shock massfield reverging not adding.//112.2215

float Break_04 = particlefield / S1; //(shockwave broken totalshock perfield)===DX)
```

```
    Debug.Log(Break_04);//1.229287

    float ObjectscatterM = ((EMV_Object / EMV_F) * M);

    Debug.Log("Scattered EMV OBJECT >  " + ObjectscatterM);

    //3.466517..

    float quanticbreakdown_Objectscatter = ObjectscatterM / Quanta;

    Debug.Log("No. quatas scattered  >  " + quanticbreakdown_Objectscatter);

    //0.6489 Mev. rest mass for sharpnells.

    float AfterShockmass = (EMV_F / (2 * Mathf.PI)) * M;

    Debug.Log("Aftershock mass  >  " + AfterShockmass);

    // shock acting on a particle. exceeding to 67+. 3.971947
```

2.2.11 Decay

Here the rest are pretty self explanatory as their properties are calculated chronologically from the top, of finding the basic parameters of quantum phenomenons. and you would see that the space and the frequency ad the relation ship tat are found in the middle of this calculation, are very related. the decay rate can be proven with Neutronic decay's rate. and other decay rate. if fact if you say that electrons decay that has been matched as well wt the same theory.

2.2.11.1 Quantized Structure. Particle

we have calculated the force the pressure the mass it ca create with the experiments and different levels of forces and got to know the vision of the architecture of the quantum. but there are yet phenomenons that need to be calculated before we jump into equations of particles found. to prove that this theory is on its track without a delay that it defines the formation and nature of the different aspect of the quantum or atom it is therefore calculated.

code

```
    /// Quantic decay

    float Decay_rate = EQ_frequency / timegap_dt;

    Debug.Log("Decay rate" + Decay_rate);//0.0600845456
```

```
        float QGap = Quanta / Decay_rate;

        Debug.Log("Qgap  " + QGap);//88.9434

        float Qgap_influence = QGap * M * Probabilty_space * Qubicmass;

        Debug.Log("Qgapinfluence  " + Qgap_influence);

        //1270.448

        float Object_Decay = objectmass / Qgap_influence;

        Debug.Log("Object decay " + Object_Decay);//0.9628749

  //scalar rest ratio after collision. multiply it with the after collision calculations, where M=0;

        float bosonic_decay = Exp_f / Decay_rate;

        Debug.Log("bosonic decay  " + bosonic_decay);//457.3143

        float Forcedecay_r = totalforce / bosonic_decay;

        Debug.Log("Force_decay  " + Forcedecay_r);//2.825096

        float TotalForce_Static = totalforce / Forcedecay_r;

        Debug.Log("totalforce  " + Totalforce);//89.418
```

2.2.11.2 Grid

```
float pergrid_prpagation_prob = 1f + (3f * 5f) + (3f * 5f * 7f);

        Debug.Log(pergrid_prpagation_prob);

        float probabilty_propagation = pergrid_prpagation_prob / Dx;

        Debug.Log(probabilty_propagation);

        //94.20.

        float potetial_grid = (TotalForce_Static * Probabilty_space) / total_potential_E;

        Debug.Log("potetial grid  " + potetial_grid);
```

```
float Total_grid_section = totalforce / potetial_grid;

Debug.Log("Totalgrid section  " + Total_grid_section);

//16.25168

float DDX = (Dx * Total_grid_section) / (dt * 4);

Debug.Log("area of DX >DDX   >>  " + DDX);

//0.6147448

float E_gradient = total_potential_E / DDX;

Debug.Log(E_gradient);//18.36869
```

2.2.12 Omega Baryonic limit

- Up
- Down

the omega particle is a strange particle. It is to be understood that you don't need to name 1.12 and 1.13 if they are consistently appearing at the same place with different values. Specially when you are dealing with a standard value or singular certainty that all are made of E. then you must call the decay or decreasing E of the particle lifetime and see the birth of a new one, a transposition. and the explanations from xi baryons to lambda baryons are all omega baryonic particle.

2.2.12.1 Upper limit

//ratio of EM Charge in space reltive to time space;its not effected by G. it is the tensionof the particle boundary limit.

```
//

float DX_ratio = Total_grid_section / Dx;

Debug.Log(DX_ratio);//12.6533

float Depth = 1;

float step = 0;

for (int i = 0; i < 6; i++)
```

```
{
  if (Depth > 0)
  {
    Depth = (Total_grid_section - DX_ratio) * probabilty_propagation;
    DX_ratio++;
    Debug.Log(Depth);
    step++;
  //338.9988
  //244.7901
  //150.5815
  //56.37277
  //-37.83581
  }
}
Debug.Log(step);//5
//
```

2.2.12.2 Scale

```
float totallength_Dx_space = Total_grid_section * step;
      D
ebug.Log(totallength_Dx_space);
      //81.25842 indexes for DX. the value then becomes
      float DX_value = totallength_Dx_space * Dx;
```

```
    Debug.Log(DX_value);

    //104.3669.

    float ratio_Index_toDX = DX_value / Total_grid_section;

    Debug.Log(ratio_Index_toDX);

    //6.421914.

    float area_Dx = Mathf.Sqrt(totallength_Dx_space) * 3f;

    Debug.Log(area_Dx);//27.04304

    float reletivsize = area_Dx / Total_grid_section;

    Debug.Log(reletivsize);

    //1.664014

    float Angstrum = 2 * (totallength_Dx_space / 2) * Mathf.PI;

    Debug.Log("Angstrum  >> " + Angstrum);//255.2809
```

2.2.12.3 Lower Limit

```
float size_dx = reletivsize / delta;

    Debug.Log("max size_DX  " + size_dx);//1.008714

    float totalE_capacity = (E / DX_value);

    Debug.Log(totalE_capacity);

    //0.022144522

    //now if dx value is linear potential. 22 is a circle. string theroy may be correct. but its not a coil in stickng to DX.

    //but yes spheric.

    float DX_inegration = (Total_grid_section / Dx);

    Debug.Log(DX_inegration);//12.6533
```

```
float Omega_particles = 0;

float Chargedepth = 1;

for (int i = 0; i < 6; i++)

{

    Chargedepth = (Total_grid_section - DX_inegration);

    Omega_particles = (Chargedepth * E * M * EMvgenaraterate);

    DX_inegration--;

    Debug.Log(Omega_particles);

//336.0432

//429.4305

//522.8179

//616.2052

//709.5925

//802.9798

}
```

2.2.12.4 Scale

```
Debug.Log(DX_inegration);//6.653302

float Particle_size_E = Empty_probablity_space * Angstrum;//2451.84

float Particle_Size = Probabilty_space * Angstrum;//501.102

Debug.Log(Particle_Size + "    " + Particle_size_E);

float total_particle_area = Particle_Size + Particle_size_E;

Debug.Log(total_particle_area);//2952.942
```

```
    float quantic_Size = Quantum * Angstrum;

    Debug.Log(quantic_Size);//442.1594

    Debug.Log(quantic_Size / TotalEmv);//41.97493

    float gridsize = (1 / Angstrum);

    Debug.Log(gridsize);

    //0.003917254

    //INDEX to Angstrum.

    float X = gridsize / Total_grid_section;

    Debug.Log("X   > " + X);//0.0002410368

    float Planksize = gridsize / (16f * 16f);

    Debug.Log("Plank X >> " + Planksize);//1.530177 E -05
```

2.2.13 DX Stress

```
float stress_grid = Totalstressfield / Total_grid_section;

    Debug.Log("Stress_grid" + stress_grid);//1.199231

    float Electric_decay = ((EF_peak / Decay_rate) * Electron_purge) / Electropotentialfiled;

    Debug.Log(Electric_decay);//0.1319533

    float static_electric_charge = Electron_purge / Electric_decay;

    Debug.Log("static E_charge" + static_electric_charge);//6304.458

    float propagation_space = probabilty_propagation / Total_grid_section;

    Debug.Log(propagation_space);

    //5.796856

    // given any point the permutation steps for propagating in space for bouded force is 5.7 dx.
```

```
float total_stress_cell = propagation_space * TotalForce_Static;

Debug.Log(total_stress_cell);

//2650.985

float Total_stress_field = total_stress_cell * Dx * Total_grid_section;

Debug.Log(Total_stress_field);

//55335.04

float stressload = Total_stress_field * 3f;

Debug.Log(stressload);

//166005.1;

float qubictension = stressload / Dx;

Debug.Log(qubictension);

//129248.9

float E_tensor = qubictension * E;

Debug.Log(E_tensor);

//298723.8

float exceed_tension = 3064888f / E_tensor;

Debug.Log(exceed_tension);

float stressDX = Dx * exceed_tension;

Debug.Log(stressDX);

//13.17769

float E_Stress = stressload / stressDX;

//Exceed E per DX
```

```
Debug.Log("E_stress" + E_Stress);

float EF_field = (((Dx * probabilty_propagation) * E_Stress) / dt * M) / (M * E * Empty_probablity_space);

Debug.Log("E_field" + EF_field);

//8089.333

float External_stress = EF_field / Probabilty_space;

Debug.Log(External_stress);

float ratio_spinstress = EF_field / External_stress;

Debug.Log(ratio_spinstress);

float total_E_stress = (EF_field / ratio_spinstress) + (External_stress / ratio_spinstress);

Debug.Log("TotalE_stress  " + total_E_stress);//

Debug.Log(total_E_stress);

//6220.43

float Perdx_Estress = (total_E_stress / Dx);

Debug.Log("perdx E_stress" + Perdx_Estress);//
```

2.3 Gravity

The Gravity is a strange thing. for Quantic gravity we can say that it is a field none the less. and that it does not attract but bends the field of mass and makes it a particle. if G wasn't there then the mass would have no center. a quantic G is similar to the G of the macro state that they are both Neutronic properties. but in this case the Neutronic matter is the total force to the length of its dimension that bends the mass. A force of Energy that si propagating through the mass field. a more indepth analysis is given in the publication of Soalar sytem and quatum formuisation.

Code :

```
float Total_E_volume = Mathf.Sqrt(E_gradient * Total_grid_section);

Debug.Log(Total_E_volume);
```

```
        float GE = TotalForce_Static * DDX / Total_E_volume;

        Debug.Log("GE" + GE);

        //16.27GE per index is 1.

        Debug.Log(delta / GE);

        Debug.Log(Total_E_volume / GE);

        float EmptyspaceG = (total_potential_E / GE) * Empty_probablity_space;

        Debug.Log(EmptyspaceG);

        //Perfect match. 6.6539

        float QuanticG = ((total_potential_E / GE) * QtoF) / Probabilty_space;

        Debug.Log("Particle G" + QuanticG);

        //66.57. per quantic beam G > 100%.

        Debug.Log(QuanticG / EmptyspaceG);//0.100126

        float MasstoE = (((((M * E) * ((Dx / dt) * GE)) / (Total_E_volume / TotalCharge)) * CbyF));

        Debug.Log("MAss to E< >> " + MasstoE);
```

2.4 Waves

the Plank scale is very important as it is a proven phenomenon of electro magnetic resonance. the wave frequency has ben calculated to its own method and it is a match with that of the plank scale on an angstrom. thus it furthers the effort to continue this calculation. it has been used to define a finer scale. the scale of gravity and the scale to the limit we can reach. theoretically.

code :

```
        float Plank_Scale = (Planksize / E);

        Debug.Log(" Plank_Scale " + Plank_Scale);//6.620623 e-06

        float wavelength = Plank_Scale / E;

        Debug.Log("Wavelength " + wavelength);//2.864547 e-06
        float Planktime = Planksize * dt;

        Debug.Log("Planktime second  >> " + Planktime);//0.0001298912

        float Px = Planksize * dt * dt;

        Debug.Log("PX >  " + Px);//0.011026
```

2.5 Particles

2.5.1 Fermions / Hadrons

Proton

```
float Proton = EM * rateEtoP * (totalinternalstressmass / TotalEmv) * (Total_E_volume / GE);

        Debug.Log(Proton);

        //a charged mass in the particle dimesion is Proton.or the amount f charge neutron can carry.

        Debug.Log("Proton  >>  " + Proton / Csqr + "  MEV/C^2");//938.8964 mev/c2
```

Neutron

```
float Neutron = (TotalEmv * Qubiccharge) / (Electron_Field / GE);

        Debug.Log(Neutron);

        Debug.Log("Neutron  >> " + Neutron / Csqr + "  MEV/C^2");//939.5948

                                        //neutron seems to be the solid chrae of object that is in everywhere.

                                        //when oyu hit oyu hit neutrons. it must be a  massive particle of bounded g.

                                        // its almost the same size as proton. 84.4439280408. the diffrence si .12.

                                        //oyu need a very special ratio of chrage vs E.
```

2.5.2 Leptons : Electron : Positron

```
 //electron
```

```
float turnbulance_field = (((E * M * spinfield * timegap_dt) / ((Dx / dt) * DDX * Totalforce * Total_E_volume)) / Probabilty_space);

Debug.Log(turnbulance_field);//0.06438545

float friction = (turnbulance_field / (GE * DDX));

Debug.Log(friction);//0.06438545

Debug.Log(totalMtoE);//1.017596

float rate_catalystC = (linearpotential / E_Divergence);

Debug.Log(rate_catalystC);//1.02115

float E_deviance = Electron_Field / GE;

Debug.Log(E_deviance);//1.039455

float ratio_delta_q = Quantum / delta;

Debug.Log(ratio_delta_q);//1.049957

float E_Quantumcycle = ((((total_potential_E) / propagation_space) / (Empty_probablity_space)) * Quantum_Cycle);

Debug.Log(E_Quantumcycle);//0.7025823

float electron = (friction * E_Quantumcycle) * (totalMtoE / (Qubicmass / EmptyspaceG));//cycle friction E.

Debug.Log("Electron > " + electron / Csqr + "  EV/C^2");//0.5115378

//positron

float positron = ((turnbulance_field / Electron_Field) * part_purgex) * (E_deviance / rate_EtoD);//outwards

Debug.Log("positron >" + positron / Csqr + "  EV/C^2");//0.05111033

Debug.Log(part_purgex);

//the problem is 1.16f vs 1.167067f;The calculator you are using. and the digits change. suppose
```

```
        //its the Macintosh i'll have  acomplete diffrent value. this is visual C value.
```

2.5.3 Quarks

- Up
- Down
- Strange
- Charmed
- Bottom
- Top

Up_Quark

```
float EM_CC = ((E * M) / spaceto_E) / particle_01;

        Debug.Log("upquark  " + EM_CC / Csqr + "  MEV/C^2");//upquark!2.321176

        Debug.Log(EM_CC);
```

Down_Quark

```
 float EDivP = (E / M) / ((Probabilty_space * Electricfield) / (per_quantam_charge / E_DX));//down quark

        Debug.Log("Downquark  " + EDivP / Csqr + "  MEV/C^2");//4.819027

        Debug.Log(EDivP);
```

Strange_Quark

```
 float EMDiv = ((EMv / Quanta) / cycle_offset) * (Total_E_volume / GE);

        Debug.Log("Strangequark  " + EMDiv / Csqr + "  MEV/C^2");//strange quark//101.1635

        Debug.Log(EMDiv);
```

Charmed_Quark

```
float EM_C = E * M * (Quanta * dt); //charm quark

        Debug.Log(EM_C);

        Debug.Log("charmquark  " + EM_C / Csqr + "  MEV/C^2");//1271.993
```

bottom_Quark

```
float EM_bot = (((E * M) * QtoF) / Dx) * breakexceed_spc;

            Debug.Log(EM_bot);

            Debug.Log("bottomquark  " + EM_bot / Csqr + "  MEV/C^2");//bottomquark//4163.585
```

Top_Quark

```
   float EM_Top = (((E * M) * TotalCharge / Probabilty_space) * (Dimension * dt)) * totalMtoE;

            Debug.Log(EM_Top);

            Debug.Log("Topquark  " + EM_Top / Csqr + "  MEV/C^2");//topqurak//173959.5
```

2.5.4 Mesons

- Phi-Meson
- Pi-Meson

Phi Meson

```
   float Phi_Meson = ((probabilty_propagation / E_deviance) * breakexceed_spc);

            Debug.Log(Phi_Meson);

            Debug.Log("Phi_meson" + Phi_Meson / Csqr + "  EV/C^2");//1020.908
```

Pi Meson.

```
     float Pi_mason = (((Dx * TotalCharge * E) / (dt * DDX)) / Total_grid_section) / rate_catalystC;

            Debug.Log(Pi_mason);

            Debug.Log("Pi_Mason  > " + Pi_mason / Csqr + "  EV/C^2"); //134.362
```

2.5.5 Annyons

- Kaon
- Gluon
- Muon
- Tau
- Higg
- Photon

- **Kaon**

```
 float KaoN = (TotalForce_Static / (exceed_tension)) * (GE / Total_grid_section);

          Debug.Log(KaoN);

          Debug.Log("KaoN > " + KaoN / Csqr + "  EV/C^2”);//496.537
```

- **Gluon**

```
 float Gluon = (Quantum * (Dx / dt) * GE) / (total_potential_E * Mathf.Pow(E_DX, 2)) * totalMtoE * breakexceed_spc;

          Debug.Log("Gluon_G >> " + Gluon / Csqr + "  EV/C^2");//energy will steady at DX.

                                              //Debug.Log(Quantum);

                                              //OK. 1.336735

          float Gluon_wavelength = Gluon * wavelength;

          Debug.Log("gluon_wave " + Gluon_wavelength);//3.441461 e-07
```

- **Muon**

```
  float MUON = (Qubicmass * Qfreq * current_E) * rate_EtoD;//105.3967

          Debug.Log(MUON);

          Debug.Log("Muon" + MUON / Csqr + "  EV/C^2”);//105.3614
```

- **Tau**

```
float tau = ((E * purge) * DDX * (Dx * dt * dt) * (total_potential_E / (exceed_tension))) * breakexceed_spc;

          Debug.Log(tau);

          Debug.Log("Tau  >> " + tau / Csqr + "  EV/C^2”);//1776.832
```

- **Higgs**

```
     float Higgs = ((Ex_cube / feedback) / rateEtoP) * totalMtoE;

          Debug.Log(Higgs);
```

```
Debug.Log("Higgs_particle" + Higgs / Csqr + "  MEV/C^2”);//1.165976
```

- **Photon**

```
Debug.Log("Photon > " + quantafrequency / Csqr + "EV/C^2");//36.02711

float Photon_charge = ((quantafrequency * Speed_quantum) / (total_potential_E * TotalCharge)) / Coulmb;

Debug.Log("Photon_C in Meter" + ((0.00000000001f * Photon_charge) / (Angstrum) + "Meter"));

//1.169719 e_17 meter
```

2.5.6 Neutrinos

- Electric
- Tau
- Muon
- Charged

```
float Neutrino_E = (wavelength * 2f) * (Totalforce / 2f) * (forceofcharge / TotalEmv) * Quanta;

float Neutrino_T = (wavelength * 2f) * (Totalforce / 2f) * (forceofcharge / TotalEmv) * Q_Circle;

float Neutrino_M = (wavelength * 2f) * (Totalforce / 2f) * (forceofcharge / TotalEmv) * Qfreq;

float Neutrino_C = (wavelength * 2f) * (Totalforce * 2f) * (forceofcharge / TotalEmv) * current_E;

Debug.Log("E_netrino " + Neutrino_E + " EV”);//   0.02665916

Debug.Log("T_netrino " + Neutrino_T + " EV”);//.   0.01058267

Debug.Log("M_netrino " + Neutrino_M + " EV”);//. 0.006170115

Debug.Log("C_netrino " + Neutrino_C + " EV”);//.  0.02278326

Debug.Log("Neutrino " + (Neutrino_E + Neutrino_M + Neutrino_T + Neutrino_C) + " EV”);//. 0.0661952
```

The Neutrinos are a wave carrying force or a force wave in energy equivalent forces where it is needed to define that E converts to force. Which One will find in the conversion equations. the neutrino family although theorized and found to be three to be of the type of wave it is it can be defined in four categories. the right handed diagonal. the left handed diagonal. the straight or horizontal and the vertical one. it is a simple method to define the propagation of wave but in the sense that if a force propagates how can it be in a form of its action. such as the vector not is the charged neutrino which is straight and horizontal and carries the maximum amount of force for the neutrinos. the Electro magnetic wave whether it interacts with the force is a matter of assuming that the sound or the force vibration frequency or the disturbance can be related to light or a laser. If indeed the laser or photopic material has any interaction with the neutrinos then the force can be effected with other types of force type waves.

2.5.7 Baryons

All Baryons Emerge From The Properties Of The Omega Baryonic Limits. Both The Upper And Lower Limit. See The Omega Baryonic Limit In Previous Explanations.

- Omega
- Lambda
- Sigma

2.5.7.1 Omega

```
float grid_DX = Total_grid_section / Dx;//

Debug.Log(grid_DX);

float D = 0;

for (int i = 0; i < 3; i++)

{

    D = (Total_grid_section - grid_DX) * probabilty_propagation;

    grid_DX++;

    //Debug.Log(Depth);

}

float omegaminus = D;

Debug.Log(omegaminus);

Debug.Log("Omega _  " + omegaminus / Csqr + "  EV/C^2”);//1675.445
```

2.5.7.2 Lambda

```
Debug.Log("++ " + charge / total_potential_E);//1.052369
Debug.Log("++ " + charge / cycle_shift);//0.5773723
Debug.Log("++ " + cycle_shift / total_potential_E);//1.822686
float Lambdanot = (Totalforce * (total_potential_E / Q_functionspace)) / rate_EtoD;
Debug.Log(Lambdanot);
Debug.Log("Lambda 0 > " + Lambdanot / Csqr + "  EV/C^2");//1115.638
float LambdaPlus = forceofcharge * (GE / Total_grid_section);
Debug.Log(LambdaPlus);
Debug.Log("Lambda + > " + LambdaPlus / Csqr + "  EV/C^2");//2286.379
Debug.Log(M_perE / Q_functionspace);
float Lambda_b = (QtoF * Mathf.Pow((Converged_E), 2)) / 1.003f;
Debug.Log(Mathf.Pow(Converged_E, 2f));//1.033729
Debug.Log("Lambda Bot > " + Lambda_b / Csqr + "  EV/C^2");//5621.304
```

2.5.7.3 Cascades

Xi

```
float DX_grid = (Total_grid_section / Dx);
Debug.Log(DX_grid);
float omega_depth = 0;
float omegathree = 0;
for (int i = 0; i < 3; i++)
{
   omega_depth = (Total_grid_section - DX_grid);
   omegathree = (omega_depth * E * M * EMvgenaraterate);
   DX_grid--;

   //Debug.Log(omegathree);

}

float XI_bot = omegathree / E_perCycle;

Debug.Log(XI_bot);

Debug.Log("XI Bottom >  " + XI_bot / Csqr + "  MEV/C^2”);//5792.332
```

2.5.8 Bosons

Bosons are Force fields. In another way, they don’t exist except for a reaction chain in the experiments. But its a property of nature how forces behave. And in the same circumstance of Environment they are at presence.

```
float Higgsboson = ((EtoF * Excubeparticlefield) / Dimension) * Gluon;

Debug.Log(Higgsboson);

Debug.Log("Higgsboson >  " + Higgsboson / Csqr + "  EV/C^2");//125100.7
```

The W,Z,S and K bosons its description only exists in the break tests. either it is a stress mass boson or a action mass boson. depends which point of view you are looking at. a particle hitting and a particle disintegrating is not the same. it is there fore not calculated as it is not understood by what it means.W boson may very well be the F-boson describe here. The boson could very well be the action boson. And vice versa until it thoroughly understood which test gives which results. A definition of these force s are not understood as a force wave cannot be defined in terms of proton transposed into w boson and disintegrating into a lambda. Or it means a z boson hits a proton which disintegrates a w boson and a lambda. The difference is this puts the W and Z not as a property of the particle but a result of hitting the proton with Z boson mass. The experiment will show that there are other massive forces involved such as the stress limit of the particle and the pressure field created by the collision. They are both introduced as chronologically S and K bosons. These can be found in the stress and brea tests section.

2.6 Antiparticles

2.6.1 Anti-neutron

```
// float Neutron = (TotalEmv * Qubiccharge)/(Electron_Field / GE);

     float antineutron = ((MasstoE * rate_EtoD) / Converged_E) / (GE / Total_grid_section);
```

```
Debug.Log(E_Divergence / per_quantam_charge);//1.006519

Debug.Log(Qubicmass / EmptyspaceG);//1.001247

Debug.Log(GE / Total_grid_section);//1.001205

Debug.Log("Anti_Neutron  " + antineutron / Csqr + "  MEV/C^2");//939.0348
```

2.7 Transpositions

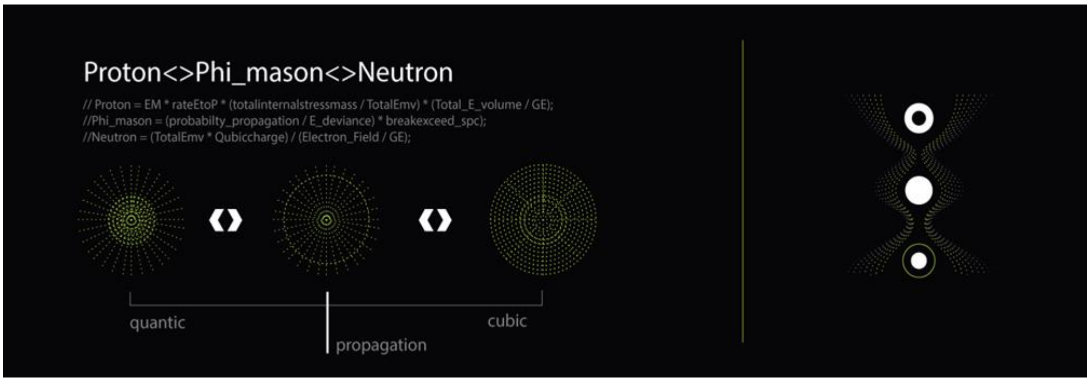
Transpositions were first Observed on experiments of the BrookHeven national Lab. They were filmed and theorized. Richard Feynman came with the Feynman diagram to research on the theory of transpositions to find out if it is an atom we are talking about, then, is it with all of its phenomenon the same. he won a Nobel prize for physics ad so the others. but the transpositions happen a bit different way in this theory of transpositions.

2.7.1 The Theoretical way

the theoretical way is not the mathematical logic of adding adjusting and the phenomenons to produce the exact same result or match match making. such as if you say three quarks is 3 kg's then a proton is 10 kgs ad there must be something causing the theory not to have defect but to call the defect another theory but realize that you were deceived by the change in values. if your equation has a linear potentiality of matter and a kg/per meter frame-rate you can see and prove that if you have 10 kgs of water you can make 5 grams of ice. but ice expands when its frozen...such and so the theoretical way is to say that there are causes and compound of phenomenons at play. they are different in this manner in your theory to match the nature of the particles position and value along with everything else. then my friend we have a theory that is correct to be true. different observations must therefore be true for all of them as you are changing the constituent of the phase that is creating this value. if the value constantly appears then it must mean that this value is special ad constant. thus it is a particle of E but with different composition as neutron ad proton are the same atom but the difference in masses makes you think that the core of atom is made out of different quarks, but the combination of this playground rule is different. the classical measurement is not the logic to quantized theories or a newtonian law of macro phenomenon. observation reseeds the theory. but you have to theories the way it is made an what it actually can be causing the to be a particle and have all these different interactions. the theory is a formula to find the omega baryonic particle by producing experiments that defined it.

2.7.2 The Practical way

the practical way is t theories with practical matters or engineering the electro dynamics with electro dynamics properties so you can deduce or reduce to practical things. the observation is not needed as you would have practical knowledge of grams and kgs of EV. And make an atom that is uniquely same. or a clone from DNA's Engineering removing the elements of levels of compounds inside the cell. or updating the E of the particle. May be if you work in Cern you can do that. t make bomb you need to be a Pyromaniac Engineer. but then you find out the mother gave birth to the clone with a different injection.



Proton<>Phi_mason<>Neutron

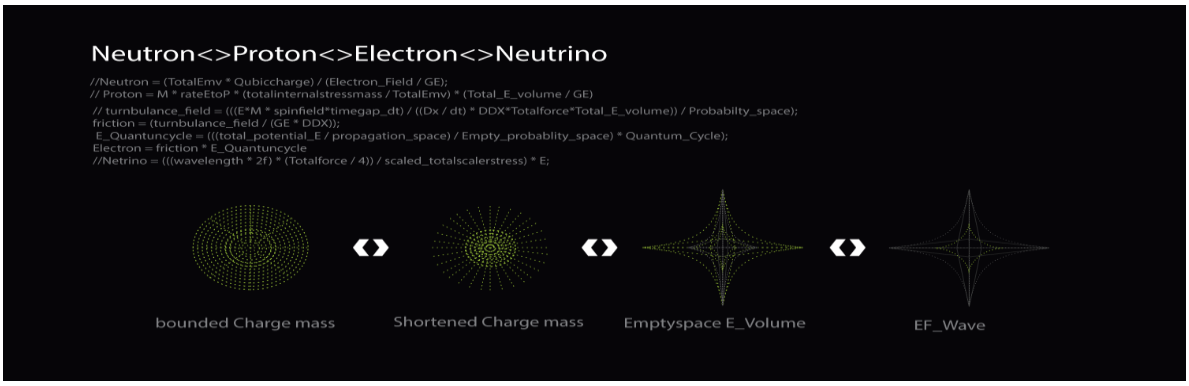
```
// Proton = EM * rateEtoP * (totalinternalstressmass / TotalEmv) * (Total_E_volume / GE);
//Phi_mason = (probabilty_propagation / E_deviance) * breakexceed_spc);
//Neutron = (TotalEmv * Qubiccharge) / (Electron_Field / GE);
```

Figure 1

2.7.3 The Mathematical way

the error in Mathematical logic is that except for the right Triangle A sqr is not equal to B sqr + C sqr . but in a philosophical way the base must be a enough to hold the a and the b pressure. the Contrasting theory of triangle is that the very anti thought about this formula created the other types of triangle based on the symmetry of the lengths. Again, another problem is with numbers. a real number decreases with squaring it, increases lowering it by root. and integer does the normal thing. such as 0.01 will increase with square. And 1.01. will increase with a square. The roots are the same. This is the inverse rule of mathematics and integers. any number below 1 and any number above one. like the omega baryonic limit. it will exceed or it will proceed. Here the space does not simply matter, neither does the particle. because its a particle of E. so E 1.2 and E 1.9 means the same thing. it will be called decaying, if the value actually decreases. Such as, if a particle is a top quark. it should decay into a up quark. or a neutron should decay into proton. Just a negative value of adding and subtracting values. in this way the C to E would be 11.33 root decays into whatever is 3.4 mev. because it’s all E. don't be confused by the theory that it has different form and function and nature of the equation. The way to go on about transposition is to squaring the values or rooting them directly. then you find out that 1.0006 square is 1.0003. it actually decreased. and thus it is a way of decaying in to other particles. if it always stops at the same point, again then it is special.

2.8 Neutronic Decay



Neutron<>Proton<>Electron<>Neutrino

```
//Neutron = (TotalEmv * Qubiccharge) / (Electron_Field / GE);
// Proton = M * rateEtoP * (totalinternalstressmass / TotalEmv) * (Total_E_volume / GE)
 // turnbulance_field = (((E*M * spinfield*timegap_dt) / ((Dx / dt) * DDX*Totalforce*Total_E_volume)) / Probabilty_space);
 friction = (turnbulance_field / (GE * DDX));
  E_Quantuncycle = (((total_potential_E / propagation_space) / Empty_probablity_space) * Quantum_Cycle);
 Electron = friction * E_Quantuncycle
 //Netrino = (((wavelength * 2f) * (Totalforce / 4)) / scaled_totalscalerstress) * E;
```

Figure 2

The neuronic deca is calculated rom the decay of the quantum properties. It is there fore comprehendible what is the reason it decay's and how it it is so.

Neutronic decay

```
MasstoE = (((((M*E)*((Dx/dt)*GE))/(Total_E_volume/TotalCharge))*CbyF))
N_decay = (MasstoE * Decay_rate*Neutron)/(E_Lifetime*Quantum_Cycle)*Qspace_dimension;
Debug.Log(N_decay);//900.dt.sec. dx is E-9 data 899. ~ 15 minutes.
```

Figure 3

```
 //lifetime.5.66 for E.

     //M per E. conversion is

     //mass decay rate/ is free Energy.

     //per eenrgy genarationrate *DT is energy cycle per theta.

     //per theta cycle is lifetime of a quantum cycle.

     Debug.Log(MasstoE);

     float N_decay = (MasstoE * Decay_rate * Neutron) / (E_Lifetime * Quantum_Cycle) * Qspace_dimension;

     Debug.Log(N_decay);//900.dt.sec. dx is E-9 data 899. ~ 15 minutes.
```

2.9 Electrolytes

electrolyte are supposed to be wave particles or the particle that makes the wave from E. it is self theorised but the waved of disturbance must be a wave of particles like photons. then either it is true that the waves are made out of the space phenomenons like the integer space or that after the phenomenon exists its quantic shape becomes a wave and then it gravitates to mass or form. and then function an repeat the bonding to macro level to repeat the bonding to super micro level. and more and more bonding parameters arrive when you are gravitating towards the blackhole. it is a charge calculated from the theory as it occurred in a value trying to find the quantic value of electron, And. it is 16 mev. And it is bigger than the electron (if there is any !). it is the electrolyte or the beams of the atom.

```
float Electrolite = (((((Mathf.Pow(electron, 2)) / Total_E_volume) + totalinternalstressmass) / ((Dx / dt) * GE)) / total_potential_E);

        Debug.Log(Electrolite);

        Debug.Log("Electrolite > " + Electrolite / Csqr + " EV/C^2"); //13.29582 ///

        float Electrolyte_charge = ((Electrolite * Speed_quantum) / (total_potential_E * TotalCharge)) / Coulmb;

        Debug.Log("Electrolyte in Meter" + ((0.00000000001f * Electrolyte_charge) / (Angstrum) + "Meter"));

        //4.316853 e-18 meter
```

2.10 Theron

```
// Universal Properties

        //2019.jul4

        float Fr1 = (((wavelength * 2f) * (Totalforce / 4)) / scaled_totalscalerstress) * E;

        float Fr2 = (((wavelength * 2f) * (Totalforce / 4)) / scaled_totalscalerstress) * M;

        float Fr3 = (((wavelength * 2f) * (Totalforce / 4)) / scaled_totalscalerstress) * Qfreq;

        float Fr4 = (((wavelength * 2f) * (Totalforce / 4)) / scaled_totalscalerstress) * GE;

        Debug.Log("E_wave " + Fr1);//7.321257 e-06

        Debug.Log("M_wave " + Fr2);//3.455431 e-06

        Debug.Log("Q_Wave " + Fr3);//3.916295 e-06

        Debug.Log("G_Wave " + Fr4);//5.154234 e-05

        float omega_variant_01 = Qscale * ((Fr1 + Fr2 + Fr3 + Fr4) * 10000f);

        Debug.Log("omega variant  " + omega_variant_01); //22.23543

        float Kaon_U = omega_variant_01 / 5f;

        Debug.Log("Kaon_U  " + Kaon_U);//4.447087
      float U = omega_variant_01 / EmptyspaceG;

      Debug.Log("Ulysis   >>" + U);//3.335954

    float Symmetrical_Constraint = Mathf.Sqrt(Mathf.Sqrt(Mathf.Sqrt((Total_E_volume / 55.5f) * ((Dx / dt) * Quanta) / (Quantum * (256f / 255.5f))))) * 4f;//((tripple root(((Sum / Eq.life)DX / dt))*quanta)/ Quatum)*(256 / 255.5);

        Debug.Log("Pi_U   " + Symmetrical_Constraint);//3.142143

        float Anamoly_variant = Symmetrical_Constraint / omega_variant_01;

        Debug.Log("Anamoly variant  " + Anamoly_variant);//0.1413124

        // Meta-Physical Phenomenons

        float Cy = (((Symmetrical_Constraint * U) / omega_variant_01) * 1.4f * (dt * dt)) * (BOSONIC_Field / Special_Boson) / (E * M / Csqr);

        Debug.Log(Cy);//14.89144.

                  //Anamoly variant = 1.4. the thesis. was 1/pi to 10/pi add them all up find the avge. its 1.4.

        // space

        float pX_correction = 10f;

        float tetrahydron_L = ((Mathf.Sqrt((U / Anamoly_variant) / Symmetrical_Constraint) * 3) / 7);

        Debug.Log(tetrahydron_L);//L = 1.17.4247

                       //float   L_Scale   =   (Plank_Scale/(((tetrahydron_L*3f)*2f)*((49-4)+((tetrahydron_L   * 3f)/4f))));

        float L_Scale = (Plank_Scale / 3f);

        Debug.Log("L    " + L_Scale);//2.206875 e-06

        float tetrahydron_L_scaled = tetrahydron_L * L_Scale;

        Debug.Log("Tetrahydron parameter  >>" + tetrahydron_L_scaled * 3);//7.777302 e-06

        float tether_wave = tetrahydron_L * wavelength * Px / pX_correction;
        Debug.Log(tether_wave);//3.708798E-9// the wave lentgh without the PX scale trasnsformation is 8.2 as well like

                        //thelatter. i guess i was born in avery fucking special time.

        float tether_energy = totalE_capacity * tether_wave;

        Debug.Log(tether_energy);//8.213216E -11;

                        //from this you can calculate the crystal.//tether.convoluted quanta.kiss an tell?

                        //if it is a time crystal then it has 2 sides. so multiply by 2. the L is half. so multiply by 4. the fulls are 3 sided

                        //so 4*3. 12.

        float Total_Crystal_E = (tether_energy * 6f / (TotalCharge / Totalforce));

        Debug.Log(Total_Crystal_E);//1.251738.e-10 its check pi permutation. this is the first diffrence between two integers.

                         //its 10/8. that means on a volume of 10, 80 is the volume. 20 is empty. vice versa 1.25*2 is 160. 0f 200. 40 is empty or

                         //free space. that is the convoluted freedom. dary  tetra. and band it with 80%.done.to the arc.A.

        float Singuler_Eta = (Total_Crystal_E / 3f) * 2f;

        Debug.Log("ETA  >>  " + Singuler_Eta);//8.3482 e-12
```

3.Conclusion

the atom is a visible formation of invisible phenomena of energy. to charge. charge to force. force to energy. the electric and the magnetic wave formation will give you idea what it is made of. perhaps if i had hardware knowledge i would better know to pulse a volt through a solenoid and make a radio-wave. but that is for the future. for know. this is all i could calculate till i ran out of dividing zeros in my rough page. The entire theory emerge with the idea of dividing 0/o . and i understood its all a ratio. a relativity of its properties which is nothing but void. The reason you see it because it has energy signals that matches the limit of our focus length. The reason it is solid and carbon is because its energy is per mutated to a bond and size that reflects light. beyond the plank scale electro magnetic force can't go. but its depth is beyond the baryonic limit.

3.1 To End

The entire calculation has started from elementary divisions to math magical series building techniques to the summation range to the conversion methods, finding out imagining how it is built from a architectural point of view, experimenting just like the do in Cern to find out whether the calculation is correct ad to match the reported measured data understanding what they are explaining that they understood and what it is that they calculate . What mysteries they behold in the future and theorists with their multidimensional theory. here at the end its a lattice of a sphere of very super quantic size thrones that convert into waves and waves form in to particles particles make the atom. but i beg to differ theres any quarks in proton or in neutron. just because it weighs five tons of charge does not mean it has three parts that are making it spin the other way. the charge and ate charge is yet be calculated. but the anti particles have been calculated and it seems the properties cancel each other out. without having a negative value. nothing is lost but energy released. And as we all undeniably can see with the help of a magical telescopic engineering that we are all atoms. we are all just compositions of Eta, Children of Void. Something rather than nothing but it's all a nothing that is the Void and invisible. But flesh and bones and conscience like a Social being as complex as the void itself. Remember that an integer is an idea of existence. But it does not exist.

3.2 Value table

Conversion.

C - 0.299792458 C^2 - 0.08987551813
unit Mega ~ M=1.09 mev
unit Giga = 100,000M > cern
ref: google. wikipedia. Cern. Cornell.
ref: 2014-.2018. .2019. .2020.
2014-2019. Published.
> checked and verified.
< not found reference of the value anywhere.

It has been checked over and over again as the name and their values seem to change and the baryonic and the cascades keeps on going on but the list never ends. I have decided to match only those that matters the concern, of the Cern experiment or the Fermi/ brook heaven bubble chamber experiments. And the Feynman diagrams, and other sources about decay ad transposition. teething about the time being on the right side or reverse is a strange phenomenon Feynman explained. But theoretically it is possible. you just have to be faster than that of the frame rate as you are thinking clock and your observational equipment. then one second can be at 100000. and at 10 the future could happen to change at 100000. and in one second the pressure has arrived. But nothing is faster than time. because its a constituent property of void. it is the same way for one spin time or 24 hour you can be at the same moment for 24 hour still, just so, you change the location. But after 24 hours you must face the 24 hours.

4. Results

QED. the architecture

Table 1

List	Relative ratios.	Energy	Particle
DX. 1.28 index	E > C 1.006519	E 2.31	Particle Field 6.861039
Parameter 4.00 index	E > D 1.005607	EMV Generation rate 37.04124	Particle E 17.27779
Index 16.	E > Dx 1.799486	E Convergence 1.640441	Particle C 352.0287
Quantic Scale 33.57035	E to P 9.994775	E Divergence 1.370924	Particle F 1291.957
U scale 3.335954	E > M 10.90266	E propagation	Particle G 66.57
L scale 2.206875 e-06	C > F 1.08955.	E_Field	Generation rate
Plank scale 6.620623 e-06	Q > F 188.3034	Electron Field	Decay rate 0.0600845456
	E > F 3118.814.	E Spin field 12.86179	Particle size 501.102
	C > E 1.006519	E Lifetime 1.762757	Particle space 2952.942
	M/E 10.35174	Linear Potential 1.399919	
	M > E 1.017596	Current 1.141288	
	M > E > C 137.841	Electropotential field 4.977306 e+07	
		Total EMV 10.53389	

Table 2

Quantic	Mass	Time	Wave
Quanta 5.34177 index	Mass number 1.09 mev	Index 8.4 dt	2.864547E-06
Quantum 1.73205 index		Spintime 7.296402d	
Quantic lifetime 7.425756		Quantic time 2.120481	
Quantic frequency 1.236325		Planktime 0.0001298912	
Quantic function Space 10.01398		Space time 1.251738.e-10	
Potential Quantic G 1.070992		Atomic clock 0.257131	
Mass Per Quanta 1.889283			
Per Quantum charge 1.362046			
Qubic Mass 6.673701			
Qubic Charge 8.332952			
Charge generation rate 2.745338			

Table 3

Theron	The Atom	Others
Parameter 7.777302 e-06	Empty space 2.798879	Omega variant 22.23543
Wavelength 3.708798E-9	Probability Space 1.962944	Symmetrical Constrain 3.142143
E 8.213216E -11	Neutron 939.5948	Cy 14.89144.
Eta 8.3482 e-12	Proton 938.8964	
	Electron 0.5115378	
	Positron 0.05111033	
	Mass 933.03 mev/c2	

Scale conversions

- Length Meter > Angstrum meter
- Time Second > dt = nanosecond.
- Energy E > Electron Volts
- Mass KG > M
- Charge Coulomb > 10 mev

The atomic model

This is a self initiated study and Computation to Describe the very existential Quanta. Accidentally it is the quantum mechanics what is described here. On a Luck Of sheer coincidences it is now the formalization that describes the rest of the phenomenon of scientific facts both experimental and theoretical. It was my effort to understand the quantum and the nature of the atom, its structure and the particles produced how and what. It is therefore presented as a theoretical research paper on Quantum Physics.

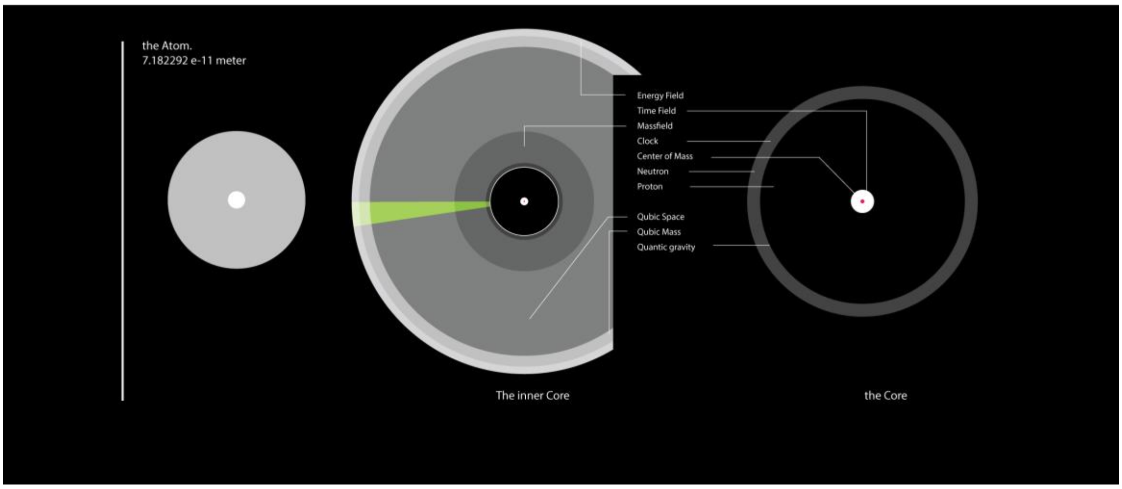


Figure 4

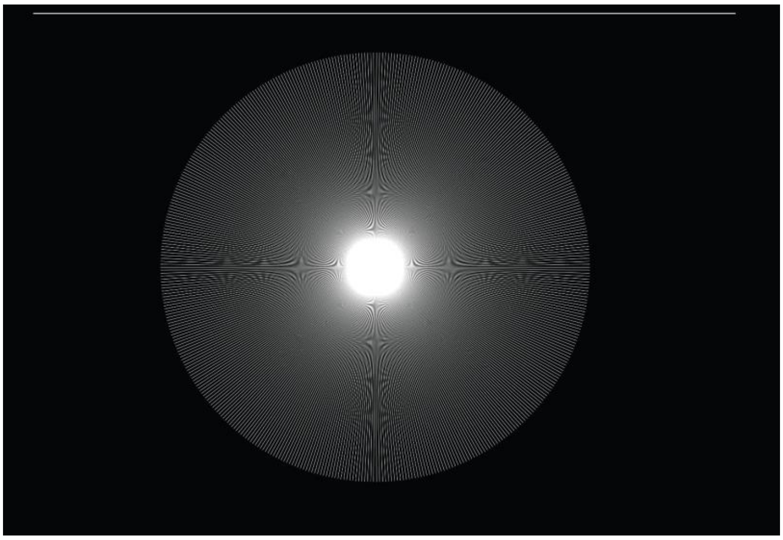


Figure 5

The above is the image of atom from a point of view of th electrolytes. It is therefore defined as a core of energy or. volume of quanta. The electro magnetic resonance can be seen with scrolling it a little or shaking the image. Which implies in visual attest that the theory of electron is that the level of energy a atom has. The atom need not to be bigger simply “Jiggle” a little faster or its electric or q frequency be faster or slower and you would see from two, 4 8 and more to 16 to 18 levels of electric level.

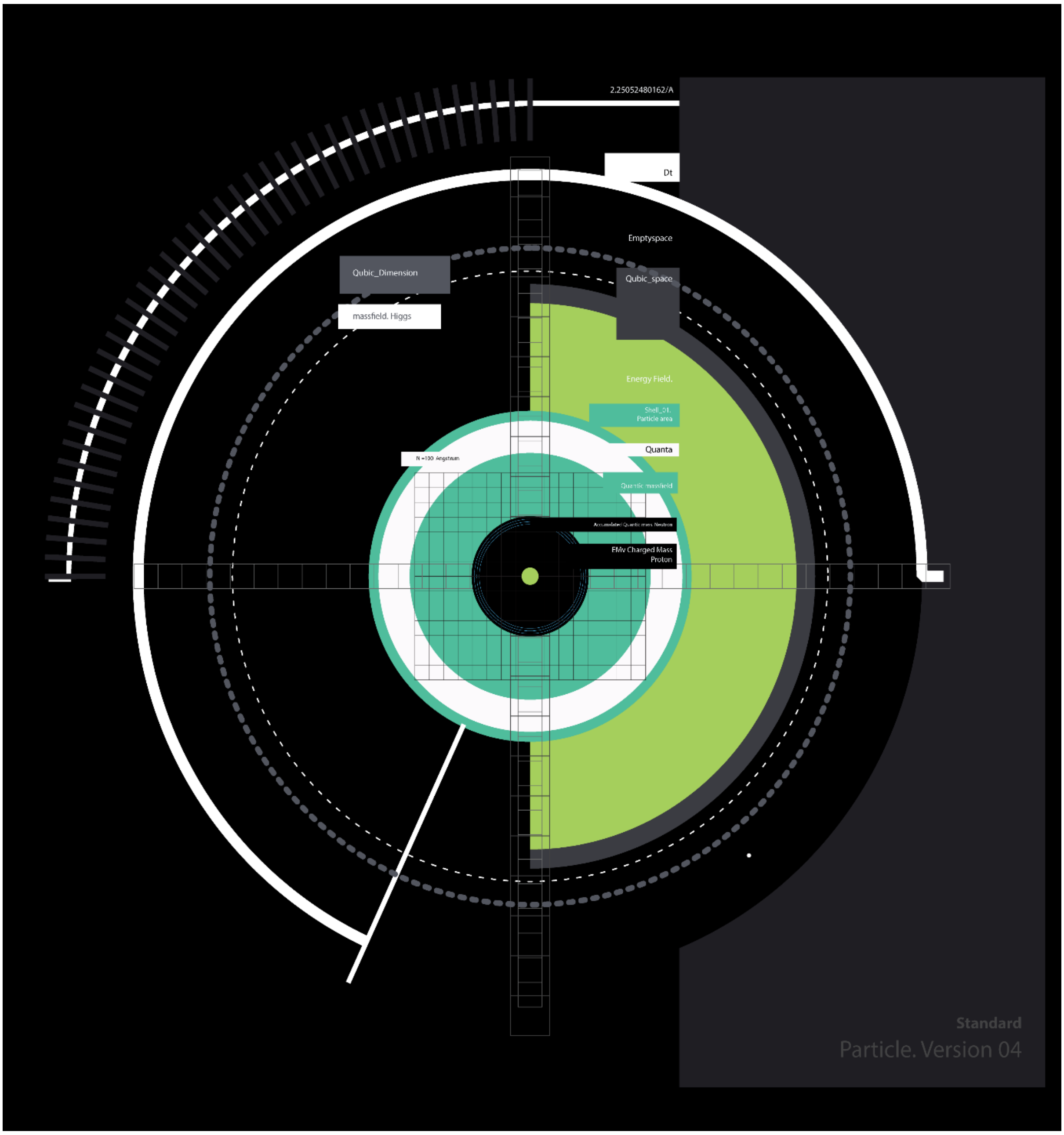


Figure 6

5. Citations

Particles and all other supporting data

5.1 World Wide Web

[1]. https://home.cern 29.02.2020

[2]. https://www.wikipedia.org. proton and neutron. 29.02.2020

[3]. https://www.wikipedia.org. Quarks. 29.02.2020

[4]. https://www.wikipedia.org. Electron and Postiron. 29.02.2020

[5]. https://www.wikipedia.org. Lamba particles. 29.02.2020

[6]. https://www.wikipedia.org. Omega baryons 29.02.2020

[7]. https://www.wikipedia.org. Xi cascades. 29.02.2020

[8]. https://www.wikipedia.org. Bosons. 29.02.2020

[9]. https://www.wikipedia.org. Neutrinos 29.02.2020

[10]. https://www.wikipedia.org. Kaon 29.02.2020

[11]. https://www.wikipedia.org. Muon 29.02.2020

[12]. https://www.wikipedia.org. Gluon 29.02.2020

[13]. https://www.wikipedia.org. phi mason 29.02.2020

[14]. https://www.wikipedia.org. pi mason 29.02.2020

[15]. https://www.wikipedia.org. Higgs boson 29.02.2020

[16]. https://www.wikipedia.org. W and Z boson. 29.02.2020

[17]. Dirac lectures you tube. 29.02.2020

[18]. https://www.wikipedia.org. Feynman diagrams 29.02.2020

[19]. https://www.wikipedia.org. Anti particles 29.02.2020

[20]. https://www.wikipedia.org. Neuronic decay 29.02.2020

[21]. Feyman Lectures youtube. 29.02.2020

*This is a self theorized calculation. It has no reference except for data to any work the has been done by any person living or dead.