THE PROOF OF X ± Y = B

Martins Kolawole Alabi

E-mail: alabikmartins@gmail.com

Abstract

This paper is about the proof of x ± y = b or x ± y = b systems at every given input number ‘n’ and a faster algorithm to the subset sum problem. The Proof of x ± y = b i.e., the proof of x ± y = b at system 1 is the proof of a mathematical method that proves something can evolve from nothing and its graph shows that the shape of the universe is a cone and this can further be mapped with an expanding universe or universes which can be used to locate the point of the big bang (a hypothetical point in space where the universe began). See No.2 at the reference list for details. However, the proof of x ± y = b at system 1 is the premise that was used here to solve the subset sum problem. Mainly, the purpose of the proof of x ± y = b is to achieve the coexistence of equal plus and minus values. The proof of x ± y = b systems are forms of x ± y = b derived from the coexistence of three quantities denoted by n, n + 1, n + 2 where n represents any given positive integers.

Keywords: input, systems, simulation, subset sum problem, algorithm

1. Definition

System ‘n’ is a positive integer (n) in which a particular x ± y = b belong.

Simulation is a valid input ‘n’ for any given system.

Simulation of x ± y = b is the system in which a valid input ‘n’ can be verified in x ± y = b.

2. Algorithm for the coexistence of three quantities denoted by n, n + 1, n + 2 where n represents any given positive integers.

Symbolic Representation

Variables A and G:

A implies Arithmetic growth mechanism +A.

G implies Geometric growth mechanism × G.

y implies pair of Spin.

Formula for the variables where n + 1 is the correlation between n and n + 2

Assuming b = n + A where result (b) also implies Index number .i.e., a set of all numbers on the number line, we have:

A = b - n

G = ((n + 1) + A) / n + 1

To generate a spin (conjugate equal and opposite value):

(n) × G minus (n) + A = +y ----------- -y

(n + 1) × G minus (n + 1) + A = 0 ----------- 0

(n + 2) × G minus (n + 2) + A = -y ----------- y

3. The problem

Unify plus and minus values.

“Prove that x ± y = b?” where b is the solution to any given number x and y for which x is number i or j at every given input number ‘n’

Given that x ± y = b.

Such that, $x - b = \pm y$ for which x is number i or j.
So that, $i - b = -y$ --------- Equation 1.
Also that, $j - b = y$ --------- Equation 2.
If it turns out that a valid input 'n' exists once and only once (which has already exists) into the equation 1 and 2. Then the solution to the problem "Prove that $x \pm y = b$?" can be quickly solved and verified by a computer. What are the formulas having at least one variable in common to create equation 1 and 2 at a time?

4. Solution

Given that $x \pm y = b$ for which x is number i or j at every given input number 'n'
So that, $i + y = b$.
Also that, $j - y = b$.
Let $i = (nb + n) / n + 1$
Let $y = (b - n) / n + 1$
Let $j = (nb + 2b - n) / n + 1$

5. The proof of x ± y = b or x ± y = b systems at every given input number 'n'

$(nb + n) / n + 1 + (b - n) / n + 1 = b$.
$(nb + 2b - n) / n + 1 - (b - n) / n + 1 = b$.
System 'n' is a positive integer (n) in which a particular $x \pm y = b$ belong. If at all system $n = -1$. Then at system -1, the $x \pm y = b$ is not applicable and it is undefined. Thus, there is undefined at infinity.

The Proof of x ± y = b at system 1 or when System n = 1

$(b + n) / 2 + (b - n) / 2 = b$.
$(3b - n) / 2 - (b - n) / 2 = b$.
System 1 can be mapped with an expanding universe or universes consisting of elementary particles that have a particle spin equal to a half-integer value (-1/2, 1/2, 3/2. etc). It is the only system used to solve the subset sum problem quickly. The only simulation that is real is the simulation of 1 or system 1.

The Proof of x ± y = b at system 2 or when System n = 2

$(2b + n) / 3 + (b - n) / 3 = b$.
$(4b - n) / 3 - (b - n) / 3 = b$.

The Proof of x ± y = b at system 3 or when System n = 3

$(3b + n) / 4 + (b - n) / 4 = b$.
$(5b - n) / 4 - (b - n) / 4 = b$.
And so on and so forth…

System $n = 0$ or system 0 is the quickest system to achieve its spin (+/- values) defined by whole integer values while system $n = 1$ or system 1 is the next fastest system to achieve its spin (+/- values) defined by half integer values.

6. Unification of all numbers

For example, let’s say result (b) is 30. Then a valid input n = all numbers. Such that we have:
(30n + n) / n + 1 + (30 - n) /n + 1 = 30.
(30n + 60 - n) / n + 1 - (30 - n) / n + 1 = 30.
Every number fit into ‘n’. Thus, the proof of x ± y = b unify all numbers.

7. A valid input (n) for system 1

Given that x ± y = b.
Such that, x - b = ±y for which x is number i or j.
So that, i - b = -y --------- Equation 1.
Also that, j - b = y --------- Equation 2.
Since, i = ((b + n) / 2) and j = ((3b - n) / 2). Therefore, we have:
((b + n) / 2) - b = -y.
((3b - n) / 2) - b = y.
Thus, a valid input ‘n’ exist into equation 1 and 2.
Therefore, x is truly i or j. Equation 1 and 2 conform.

8. What a valid input ‘n’ postulates in the proof of x ± y = b

It postulates that for every given input (n), there exist a turning point at b = n i.e. the point at which a significant change occur; a decisive moment designated as {0, 0, 0}

9. The computer program that shows that proof of x ± y = b is valid

Let take for instance system 1 where number ‘n’ is a valid input:
((b + n) / 2) - b= -y.
((3b - n) / 2) - b = y.

10. The computer program of x - b = +/-y at system 1

```
package proof;
import javax.swing.*;
import java.util.Locale;
import javax.swing.JTextArea;
public class prog
{
public static void main (String args []){
double x=0; double count=0; double N=0; double y = 0; double x1 = 0; double y1 = 0; double y2 = 0;
String input;
input = JOptionPane.showInputDialog("Enter any number for n");
        N=Double.parseDouble(input);
        JTextArea outputTextArea = new JTextArea();
for(count=-20;count<=20;count++){ //Suggested input limit of the java compiler {1.0E17, //1.0E308,
1.0E309}
x=(count+N)/2;
y=x-count;
x1=(3*(count)-(N))/2;
y1=x1-(count);
y2=y+y1;
outputTextArea.append(count + " ...... "+y+",   "+y1+",   "+y2+ "\n");
}
JOptionPane.showMessageDialog( null, outputTextArea, "", JOptionPane.INFORMATION_MESSAGE );
System.exit(0);
}}
```

11. Analysis for the simulation of the proof of x ± y = b at system 1

When a valid input n = 1, output is:

```
(b): ..... ("+/-y" values):
...(-∞)
-1........1.0, -1.0
0.........0.5, -0.5
1.........0.0, 0.0
2......... -0.5, 0.5
3......... -1.0, 1.0
...(+∞)
```

When a valid input n = 2, output is:

```
(b): ... ("+/- y" values):
...(-∞)
-1.....1.5, -1.5
0......1.0, -1.0
1......0.5, -0.5
2.......0.0, 0.0
3...... -0.5, 0.5
...(+∞)
```

When a valid input n = 3, output is:

```
(b): ... ("+/-y" values):
...(-∞)
-1.....2.0, -2.0
0......1.5, -1.5
1......1.0, -1.0
2.......0.5, -0.5
3.......0.0, 0.0
...(+∞)
```

..And so on and so forth...

12. Test validity of the computer program that shows that proof of x ± y = b is valid without changing the instance; test for validity

Input any decimal number that is not a half-integer value once and only once.

When a valid input n = 2.4, output is:

```
(b): Spin ("+/-y" values):
-1.0.......1.7, -1.7000000000000002 difference: -2.220446049250313E-16
0.0.......1.2, -1.2
1.0.......0.7, -0.7
2.0.....0.20000000000000018, -0.19999999999999996 difference: 2.220446049250313E-16
3.0....... -0.2999999999999998, 0.2999999999999998
```

When a valid input n = 0.1, output is:

```
(b): Spin ("+/-y" values):
-1.0.......0.55, -0.55
0.0.......0.05, -0.05
1.0.......-0.44999999999999996, 0.44999999999999996
2.0....... -0.95, 0.9500000000000002 difference: 2.220446049250313E-16
3.0....... -1.45, 1.4500000000000002 difference: 2.220446049250313E-16
```

Why are the different results when a decimal value that is not half-integer is input?
Since we have the same results when an integer value is input, the result must be the same when a decimal value is input. Else, there is a difference somewhere…

Reason: When a decimal value that is not a half-integer value is input once and only once, the floating point error enables us to see that what is done to achieve one value is not conversely of what is done to achieve the other value. Otherwise, despite the floating point error the result would have been the same when a decimal value is entered. Thus, the floating point numbers exist in this case of study!

Conclusion: This tells us that the plus and minus values emerge from two different sources at a given input (n). So, we are not doing the same thing but different things to achieve the plus and minus values. That is to say, the plus and minus values can coexist. Therefore, the computer program and the proof of $x \pm y = b$ is valid. Basically, what "The Proof of $x \pm y = b$" does is how one number (n), input once and only once, can give you two equal and opposite results.

Note that via a command prompt, the two opposite plus and minus values might be different when a decimal number that is not half of an integer is input. This tests its validity.

13. The graph of $x \pm y = b$ such that $x - b = +/-y$ is an asymptotic cone

For example $x \pm y = b$ at system 1.
Such that $x - b = -y$.
So that $x - b = y$.
We have:

((b + n) / 2) - b= -y.
((3b - n) / 2) - b = y.
Thus for every valid input n, the graph of b against +/-y gives us an asymptotic cone.

14. Application

(Algorithm for Subset Sum Problem)

Question: (From Wikipedia)

Consider the subset sum problem, an example of a problem that is easy to verify, but whose answer may be difficult to compute. Given a set of integers, does some nonempty subset of them sum to 0? For instance, does a subset of the set {−2, −3, 15, 14, 7, −10} add up to 0? The answer "yes, because {−2, −3, −10, 15} adds up to zero" can be quickly verified with three additions. However, there is no known algorithm to find such a subset in polynomial time.

Polynomial time means that as the complexity of the problem grows, the difficulty in solving it doesn't grow too fast. All the basic arithmetic operations (Addition, subtraction, multiplication, division, and comparison) can be done in polynomial time.

15. The procedure

(Eliminating randomness)

Randomness in this paper is referred to an act of inputting integers to the program without pattern.
Inputs distinct set of integers that sum to 0 and any two distinct integers.
This would amount to the number of input.
For example; given that, A: Set {-2, -3, 15, -10} is 6 element sets. B: Set {7, 14} is 2 element sets.
4 element sets plus 2 element sets equals 6 element sets.
Therefore, the total number of input required is 6 element sets {-2, -3, 15, 14, 7, -10} which can also be entered to the computer as input without any particular order. For instance, it can take any form like {7, -2, 15, -3, 14, -10}, {-10, -2, 14, -3, 7, 15} etc… in no particular order.
Given a set of integers does some non-empty subset of them sum to 0? For instance; Enter the following set of numbers {-2, -3, 15, 14, 7, -10}, {13, 2, 3, -27, 5, -8, 7, -6, -30, -4, 10} etc…
Check out the answer from the computer program below.
This algorithm and the computer program takes large input and output a list of distinct integers that sum to 0 and the integers are all in the set of input.
In order to generate a faster way to get your input, you can deploy a positive integer and its negative followed by any two distinct integers. i.e. {1, -1, 2, -2, 3, -3, 4, -4…9, 16}…

16. Solution

Criteria:
x ± y = b at system 1
Given that x ± y = b at system 1 for which x is number i or j.
So that, i + y = b.
Also that, j - y = b.
i = ((b + n) / 2) and j = ((3b - n) / 2).
Since 'n' is a valid input in the proof x ± y = b for which x is number i or j.
Let 'n' be our chosen criteria for selecting the subset that sum to zero.

Therefore, n is chosen from the equation i + y = b at system 1.
Such that ((b + n) / 2) + y = b.
Also, let 'n' be the subject of the formula.
And let 'i' denote each of every given input.
Such that, i + y = b becomes, input + y = b.
Since, i = ((b + n) / 2).
Thus, input = ((b + n) / 2).
n = (2 × input) - b.
Example, for 6 element sets:
Let consider q, r, s, t, u, v to be the set of input.
Let b represent the sum of all the input.
b = q + r + s + t + u + v.

Then according to the equation; n = (2 × input) - b, we have the following below:

17. The algorithm for a 6 element set eliminating randomness

```
import javax.swing.*;
import javax.swing.JTextArea;
public class Proj {
    public static void main(String[] args) {
int q=0;int r=0;int s=0;int t=0;int u=0;int v=0;int b=0; long start; long end; String input;
input = JOptionPane.showInputDialog("Follow the procedure: Choose six distinct integers or half integers
make sure four out of them sum to zero e.g., enter the set of numbers {-2, -3, 15, 14, 7, -10}");
                q = Integer.parseInt(input);
input = JOptionPane.showInputDialog("Follow the procedure: Choose six distinct integers or half integers
make sure four out of them sum to zero e.g., enter the set of numbers {-2, -3, 15, 14, 7, -10}");
                r = Integer.parseInt(input);
input = JOptionPane.showInputDialog("Follow the procedure: Choose six distinct integers or half integers
make sure four out of them sum to zero e.g., enter the set of numbers {-2, -3, 15, 14, 7, -10}");
                s = Integer.parseInt(input);
input = JOptionPane.showInputDialog("Follow the procedure: Choose six distinct integers or half integers
make sure four out of them sum to zero e.g., enter the set of numbers {-2, -3, 15, 14, 7, -10}");
                t = Integer.parseInt(input);
input = JOptionPane.showInputDialog("Follow the procedure: Choose six distinct integers or half integers
make sure four out of them sum to zero e.g., enter the set of numbers {-2, -3, 15, 14, 7, -10}");
                u = Integer.parseInt(input);
input = JOptionPane.showInputDialog("Follow the procedure: Choose six distinct integers or half integers
make sure four out of them sum to zero e.g., enter the set of numbers {-2, -3, 15, 14, 7, -10}");
                v = Integer.parseInt(input);
b=q+r+s+t+u+v;
start = System.currentTimeMillis();
//According to the equation; n = (2 × input) - b, we have:
int nn1=2*q;
int nn2=2*r;
int nn3=2*s;
int nn4=2*t;
int nn5=2*u;
int nn6=2*v;
int n1=nn1-b;
int n2=nn2-b;
int n3=nn3-b;
int n4=nn4-b;
int n5=nn5-b;
int n6=nn6-b;
//Acquisition of 0 for arguments
int c1n2=n1+n2;
int c1n3=n1+n3;
int c1n4=n1+n4;
int c1n5=n1+n5;
int c1n6=n1+n6;

int c2n3=n2+n3;
int c2n4=n2+n4;
int c2n5=n2+n5;
int c2n6=n2+n6;

int c3n4=n3+n4;
int c3n5=n3+n5;
int c3n6=n3+n6;

int c4n5=n4+n5;
int c4n6=n4+n6;

int c5n6=n5+n6;

JTextArea outputTextArea = new JTextArea();
JTextArea elseoutputTextArea = new JTextArea("No result. Please, follow procedure!");
end = System.currentTimeMillis();
//Permutation of the entire element sets reference to 0
if (c1n2==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+s +", "+ t +", "+ u +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                        System.exit(0);}
if (c1n3==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+r +", "+ t +", "+ u +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c1n4==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+r +", "+ s +", "+ u +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c1n5==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+r +", "+ s +", "+ t +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c1n6==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+r +", "+ s +", "+ t +", "+ u+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output ", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c2n3==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ t +", "+ u +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c2n4==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ s +", "+ u +", "+
v+" add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c2n5==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ s +", "+ t +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output ", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c2n6==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ s +", "+ t +", "+ u+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
if (c3n4==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ r +", "+ u +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                         System.exit(0);}
```

```
if (c3n5==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ r +", "+ t +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                              System.exit(0);}
if (c3n6==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ r +", "+ t +", "+ u+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                              System.exit(0);}
if (c4n5==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ r +", "+ s +", "+ v+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                              System.exit(0);}
if (c4n6==0){outputTextArea.append("Execution time was "+(end-start)+"ms. "+q +", "+ r +", "+ s +", "+ u+"
add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                              System.exit(0);}
if (c5n6==0){outputTextArea.append("Execution time was "+(end-start)+"ms.Yes. "+q +", "+ r +", "+ s +", "+
t+" add up to zero can be quickly verified with three additions.");
JOptionPane.showMessageDialog( null, outputTextArea, "Output", JOptionPane.INFORMATION_MESSAGE
);
                              System.exit(0);}
else {JOptionPane.showMessageDialog( null, elseoutputTextArea, "You do not follow procedure!",
JOptionPane.INFORMATION_MESSAGE );
                              System.exit(0);}
  }}
```

18. The algorithm that takes a large input eliminating randomness

```
import javax.swing.*;
public class Proof {


  public static void main(String args[]) {
             double total = 0;int colum;int coum=1;
             double x,q,lo = 1;int vb;
             String input, inp;
             double t = 0;long start=0;long end=0;
                String output = "Subscript\tValue\n";
                String seval="";String seva="";String seal="";
             input = JOptionPane.showInputDialog("How many input do you want");
```

```
            vb = Integer.parseInt(input);
            double[] array = new double[vb];
            double[] arry = new double[vb];double[] n = new double[vb];double[] c1n = new
double[vb];
          for ( int counter = 0; counter < array.length; counter++ )
          {
            inp = JOptionPane.showInputDialog("Follow the procedure and enter your value:");
            q =  Double.parseDouble(inp);
                  array[ counter ] = q;
                  total += array[ counter ];
                  seval = seval + q + ", ";
                  start = System.currentTimeMillis();
          }
            for ( int counter = 0; counter < array.length; counter++ )
          {
            arry[ counter ] = 2 * array[ counter ];
            seva = seva + arry[ counter ] + ",  ";
          }
          for ( int counter = 0; counter < arry.length; counter++ )
          {
            //start = System.currentTimeMillis();
            n[ counter ] = arry[ counter ] - total;
            seal = seal + n[ counter ] + ",  ";
            if( n[ counter ]==total ){
                  String answer=" ";
                  for ( int i = 0; i < array.length; i++ ){
                        if( array[i] != n[ counter ] )
                              answer+=array[i]+" ";
                  }//end for
         JTextArea outputArea = new JTextArea();
                              end = System.currentTimeMillis();
                              long res = end - start;
                              JOptionPane.showMessageDialog( null,"Start time was:
"+start+"milliseconds . End Time was: "+end+"milliseconds . Total time was: "+res+"milliseconds . Result
is: " + answer, " ", JOptionPane.INFORMATION_MESSAGE );
                              System.exit( 0 );
                  }//end if
          }
          for ( int counter = 0; counter < n.length; counter++ )
          {
            double cou = n.length;
            double fir = n[counter];
                  for ( colum = coum; colum < cou; colum++ )
            {
                  String hj="";
                  double sec = n[colum];
                        c1n[colum] = fir + n[colum];
                        if(c1n[colum] == 0)
                        {
                              for ( int colums = 0; colums < array.length; colums++ )
                              {
                                          double cons = counter, colu = colum;
                                    if (colums != cons)
                                    {
                                                if (colums != colu)
                                                {
                                                      hj = hj + array[colums] + ", ";
                                                }}}
                              JTextArea outputArea = new JTextArea();
                              end = System.currentTimeMillis();
                              long res = end - start;
                              JOptionPane.showMessageDialog( null,"Start time was: 
"+start+"milliseconds . End Time was: "+end+"milliseconds . Total time was: "+res+"milliseconds . Result 
is: " + hj, " ", JOptionPane.INFORMATION_MESSAGE );
                              System.exit( 0 );
                        }}
            coum = coum + 1;
            String outpt = "Subscript\tValue\tColum\n";
          }
          for ( int counter = 0; counter < c1n.length; counter++ )
          {
            double chk = c1n[counter];double len = c1n.length;
                  if(chk != 0)
            {
                        JOptionPane.showMessageDialog( null,"No result. Please follow 
procedure!", "You do not follow procedure", JOptionPane.INFORMATION_MESSAGE );
```

```
                        System.exit( 0 );
  }}
      System.exit( 0 );
                  }}
```

19. In general, the proof of x ± y = b or x ± y = b systems at every given input number ‘n’

$(nb + n) / n + 1 + (b - n) / n + 1 = b.$

$(nb + 2b - n) / n + 1 - (b - n) / n + 1 = b.$

Note: System ‘n’ is a positive integer (n) in which a particular x ± y = b belong. For instance, the proof of x ± y = b at system 1 or when System n = 1.

$(b + n) / 2 + (b - n) / 2 = b.$

$(3b - n) / 2 - (b - n) / 2 = b.$

20. References

[1] P versus NP problem- Wikipedia.

[2] Martins Kolawole Alabi. Artificial Intelligence AI Choice Mechanism
Hypothesis of a Mathematical Method that Describes Quantum Numbers, Quantum Entanglement And Expanding Universe or Universes Proving That Something can evolve from Nothing (.i.e. numbers evolving from non-existent illusion) Vol 7, (2012) Advances in Physics Theories and Applications, IISTE.
http://www.iiste.org/Journals/index.php/APTA/article/view/2155