Ternary Mathematics and 3D Placement of Logical Elements Justification

Ruslan Pozinkevych

Faculty of Informations Technologies and Mathematics, The Eastern European National University, 43021, Lutsk, Potapov str.9, Ukraine.

Author’s contribution

The sole author designed, analyzed, interpreted and prepared the manuscript.

ABSTRACT

Aims/Objectives: The research presented in the following application aims to prove use of Ternary Maths for calculating machines and to simplify the process of calculating. In it we will try to justify the use of triplets and describe how it works.

An earlier research presented in “Logical Principles in Ternary Mathematics” [1,2,3] shows that we can transit from one expression of a number such as a “component form” to another, e.g. a decimal, or still another, that is it’s vector form [4]. The aim of our further research is to explain why we associate Triplets of numbers in such choice {-1,0,1} and not the numbers 1,2,3 for example, or a set {1,2,3}. The explanation seems obvious as a set of decimal numbers consists of 10 entries not 3. At the same time we have to prove that the mentioned set of triplets is a unique and the only one to be used as a Ternary Set or a base, as we might call it, for our calculating machines.

Keywords: Base; Unique Representation; 3D placement; Logical Operators; Ternary Algebra.

1. INTRODUCTION

The idea why we even start to speak about Ternary Maths stems from the fact that any numeric system can be reduced-expanded to a minimum/maximum numbers of entries [5]. The principle can be demonstrated by the
following example. Let’s take a look at geometry. Suppose we have a quadrilateral that in its turn has 4 vertices. If we add another 2 we must receive the outcome of 6 (4+2)=6. Yet if we place the same vertices differently, we will obtain another quadrilateral again and so our addition principle will not work. Hence 4+2 ≠6. Another example from geometry. Think of the trapezium. If we add another vertex we might obtain a triangle and thus 4+1 will be equal 3. The idea of Ternary Mathematics is quite similar. We may reduce and extend the number of entries but of course their choice is not random. Our research is aimed at establishing a connection between Ternary Principle and mathematical operations (addition, multiplication) for purposes of calculating [6,7].

This method will be used primarily by computing systems where logical elements are placed on xy, yz and zx planes. Once the approach is justified the use of Ternary Maths will serve its purpose and make it a distinctive branch of Mathematics. So what gives us a right to speak about (-1,0,1) as a unique set? Well first of all we can choose eigenvectors.

I would like to begin with the analysis of Ternary addition and take a look at the entries table.

We have managed to construct Truth Tables for \((T,T,F)\) entries. Here they are:

2. METHODOLOGY

Thus far the following results have been obtained: We have managed to construct Truth Tables for \((T,T,F)\) entries and here they are:

In this case the results are as follows: \((-1)+(-1)=(-1)\);

1+1=1

It’s very important to mention it because in Ternary Mathematics this very principle plays the key role [8].

It’s high time for us to present a definition of a ternary maths. So what is actually a Ternary Maths?

A Ternary maths is a matrix presentation of a number whose orthonormal basis is a set of numbers \((-1,0,1)\). For the time being we are going to leave our “Ternary Multiplication” Table unchanged and concentrate on our “Ternary Addition”.

### Table 1. Ternary Addition

|   | A   | B   | A ⊕ B |
|---|-----|-----|-------|
| T | T   | T   | T     |
| F | F   | F   | F     |
| T | T   | T   | T     |
| F | F   | F   | F     |
| T | 0   | T   | 0     |
| F | 0   | F   | 0     |
| T | 0   | T   | 0     |
| F | 0   | F   | 0     |

*For the sake of propriety one should mention that Ternary addition works very much like addition of numbers \((-1,0,1)\) with an exception that, when we have negative one plus negative one it equals negative one. Whereas positive one plus positive one equals positive one.*

### Table 2. Ternary Multiplication

|   | A   | B   | A*B  |
|---|-----|-----|------|
| T | T   | T   | T    |
| T | F   | F   | T    |
| F | T   | F   | F    |
| T | 0   | T   | 0    |
| F | 0   | F   | 0    |
| T | 0   | T   | 0    |
| F | 0   | F   | 0    |

For the time being we are going to leave our “Ternary Multiplication” Table unchanged and concentrate on our “Ternary Addition.”
Let's group elements of the "Ternary Addition" results column into a
\[
\begin{pmatrix}
1 & z & 1 \\
x & y & 1
\end{pmatrix}
\]
which is a proper orthogonal matrix the determinant of which is equal 1. Its very important that \(\Delta \neq 0\) and that \(\Delta = 1\).

Firstly because if matrix is orthogonal that means that the entries \(x, y, z\) are orthonormal and thus are linearly independent and do not lie on the plane [9]. On the other hand we can always present this matrix in the form:

\[
abc = \begin{pmatrix}
a_1 \\
b_1 \\
c_1
\end{pmatrix} = \begin{pmatrix}
a_2 \\
b_2 \\
c_2
\end{pmatrix} = \begin{pmatrix}
a_3 \\
b_3 \\
c_3
\end{pmatrix}
\]

Where:

\[
a = a_1 e_1, \\
b = b_1 e_1 + b_2 e_2, \\
c = c_1 e_1 + c_2 e_2 + c_3 e_3.
\]

(\(e\) an eigenvector) [10,11].

This is a general approach towards 3 D placement of logical elements, the detailed explanation of which follows in our "Materials and Methods" discussion.

3. MATERIALS AND METHODS

Our next step is to find the values of \(e (e_1, e_2, e_3)\) for which the evaluation will hold. To be able to do this let's remind ourselves that our matrix

\[
\begin{pmatrix}
a_1 \\
b_1 \\
c_1
\end{pmatrix} = \begin{pmatrix}
a_2 \\
b_2 \\
c_2
\end{pmatrix} = \begin{pmatrix}
a_3 \\
b_3 \\
c_3
\end{pmatrix}
\]

where entries \(a_1 = b_2 = c_3 = 1\);

\(b_1 c_1 = c_2 = (-1)\);

and

\(a_2 = a_3 = b_3 = 0\)

(see Table 1 Ternary Addition) is the last column of Ternary Addition entries rearranged into the 3X3 input.

Even though we arranged our entries into the 3x3 matrix, our sought after matrix is not the matrix of input but the matrix of coefficients, which is an inverse of an input matrix \(A\).

Let's call this matrix \(A^{-1}\).

It is also triangular yet has different entries altogether. They are:

\[
a_1 = b_2 = c_3 = 1; \\
b_1 = c_1 = 2; c_2 = 1;
\]

In that case we obtain \(a = b = c = 1\); and input entries of the orthonormal vectors:

\[
e_1 = 1; \\
e_2 = 0; \\
e_3 = (-1)
\]

This last part is probably the most important part in Ternary Mathematics principle. We obtained 3 orthonormal vectors which we will write in the form of a column vector [12].

\[
\begin{pmatrix}
e_1 \\
e_2 \\
e_3
\end{pmatrix} = \begin{pmatrix}
1 \\
0 \\
-1
\end{pmatrix}
\]

Now every matrix of coefficients can be written as

\[
y A^{-1}
\]

Where \(y = \begin{pmatrix}
1 \\
0 \\
-1
\end{pmatrix}\).

4. RESULTS AND DISCUSSION

Our goal was to present a unique unit [13] that would be an Eigenvalue for the Ternary Mathematics calculations. This unit has been found. It's a set of orthonormal entries \((1, 0, -1)\) with which every matrix of coefficients can be presented in the form

\[
y A^{-1}
\]

What has led to obtaining this result was a group of conversions, such as Gaussian eliminations; the fact that \(A^{-1} A = I\); as well as algebraic conversions and solving systems of linear equations.

One of the steps we made on the way to solve simultaneous equations was to convert a matrix into it's inverse \(A^{-1}\).
5. CONCLUSIONS

The idea behind was to find an orthonormal basis \( e_1, e_2, e_3 \) and place these vectors in the formula to represent \( abc \)

By solving the system of three equations we found the values of \( e_1, e_2, e_3 \) first for the matrix \( A \) and then by analogy for the matrix \( A^{-1} \). Of course they are different, therefore a special operator has been chosen for the “unique” set \( y = \left( \begin{array}{c} 1 \\ 0 \\ 0 \end{array} \right) \)

We were finally ready to come up with the first postulate of Ternary Mathematics:

“A number or a single unit of counting in the Ternary system is presented as a product of a unique set \( y \) and the matrix itself” [14].
(Pozinkevych Ruslan)

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Duerlinger J. Sullogismos and Sullogizesqai in Aristotle’s Organon. The American Journal of Philology. 1969;90(3):320-328.
2. Lukasiewicz J. Aristotle’s syllogistic from the standpoint of modern formal logic; 1951.
3. Pozinkevych R. Logical Principles in Ternary Mathematics. Asian Journal of Research in Computer Science. 2021;49-54.
4. Dolciani MP. Modern introductory analysis. In Modern introductory analysis. 1967;660-660.
5. Hilbert D, Bernays P. Grundlagen der Mathematik II; 1974.
6. Russel B. Mysticism and logic and other essays. London, UK: George Allen & Unwin; 1956.
7. Wittgenstein L. Ludwig Wittgenstein. Rowman & Littlefield; 2003.
8. Pozinkevych R. Ternary Mathematics Principles Truth Tables and Logical Operators 3 D Placement of Logical Elements Extensions of Boolean Algebra. Asian Journal of Research in Computer Science. 2020;35-38.
9. Kravchuk OM. Workshop on analytical geometry; 2013.
10. Huss-Lederman S, Jacobson EM, Johnson JR, Tsao A, Turnbull T. Implementation of Strassen's algorithm for matrix multiplication. In Supercomputing'96: Proceedings of the 1996 ACM/IEEE Conference on Supercomputing. IEEE. 1996; 32-32.
11. Postnikov MM. Analytic geometry. Geometry lectures; 2009.
12. Macbeath AM. Elementary vector algebra. Oxford: Oxford University Press; 1964.
13. Cantor G. Contributions to the founding of the theory of transfinite numbers translated, and provided with an introduction and notes, by Philip EB Jourdain; 1952.
14. Cung VD, Danjean V, Dumas JG, Gautier T, Huard G, Raffin B, Trystram D. Adaptive
and hybrid algorithms: classification and illustration on triangular system solving. In Transgressive Computing. Copias Coca, Madrid. 2006;131-148.

15. Harris D, Harris S. Digital design and computer architecture. Morgan Kaufmann; 2010.

16. Li F, Nicopoulos C, Richardson T, Xie Y, Narayanan V, Kandemir M. Design and management of 3D chip multiprocessors using network-in-memory. In 33rd International Symposium on Computer Architecture (ISCA’06). IEEE. 2006;130-141.

© 2021 Pozinkevych; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Peer-review history:
The peer review history for this paper can be accessed here: https://www.sdiarticle4.com/review-history/71745