Simple Formulas for Generating Pythagorean Quintuples

Nadav Voloch[1]

[1] IMT School of advanced studies, Lucca, Italy
voloch@post.bgu.ac.il

Abstract. This paper presents a novel and accessible method for generating Pythagorean quintuples using a single-variable algebraic approach derived from the classical Platonic sequence. While the generation of Pythagorean triples and quadruples has long been studied in number theory, the extension to quintuples remains relatively unexplored. By extending known recursive formulations and distinguishing between even and odd input cases, this work introduces a family of closed-form expressions that yield valid Pythagorean quintuples with only one input variable. Although the method does not capture all primitive quintuples, it offers an elegant and efficient framework for producing a wide variety of valid solutions. The simplicity of the formulas lends itself to educational use, symbolic verification, and computational applications. A complete Python implementation accompanies this study, capable of automating the generation process and validating results across multiple input ranges. This work lays the foundation for future research in higher-dimensional Diophantine analysis and potential cryptographic applications involving lattice-based number structures.

Keywords: Pythagorean quintuples, Pythagorean quadruples, Pythagorean triples, Number theory, Cryptography, Algebraic Geometry, Diophantine Equation.

1 Introduction

Pythagorean quintuples are sets of five positive integers (a, b, c, d, e) that satisfy the Diophantine equation $a^2 + b^2 + c^2 + d^2 = e^2$ ([1]). They represent a natural extension of the classical Pythagorean triples(a, b, c), which satisfy $a^2+b^2=c^2$, and their four-dimensional analogue, the Pythagorean quadruples (a, b, c, d), which satisfy $a^2+b^2+c^2=d^2$. While Pythagorean triples have been thoroughly studied for their foundational role in number theory and geometry, and Pythagorean quadruples have also garnered mathematical interest, the exploration of Pythagorean quintuples remains relatively underdeveloped.

The study of quintuples occupies a unique position at the intersection of number theory, algebraic geometry, and higher-dimensional Diophantine analysis. These tuples not only offer insight into solutions of quadratic forms but also carry implications for cryptography, lattice-based mathematics, and multidimensional geometry.

In Figure 1 we see a rectangular box with side lengths 3, 4, and 12 units and a space diagonal of 84 units is extended conceptually into a fourth dimension—time. In this visualization, the space diagonal no longer represents only a spatial relationship between the edges but now evolves along a timeline. The fifth component, e, signifies a point in time associated with the box's spatial configuration, which is the time unit 85. This dynamic interpretation reflects the transition from quadruples to quintuples, where the time dimension joins the three spatial dimensions, and the Pythagorean identity is extended accordingly: this visual serves as a geometric representation of a Pythagorean quintuple $(3, 4, 12, 84, 85)$, where the sum of the squares of the three edge lengths and the space diagonal equals the time unit in the box's movement in space: $3^2 + 4^2 + 12^2 + 84^2 = 7225 = 85^2$.

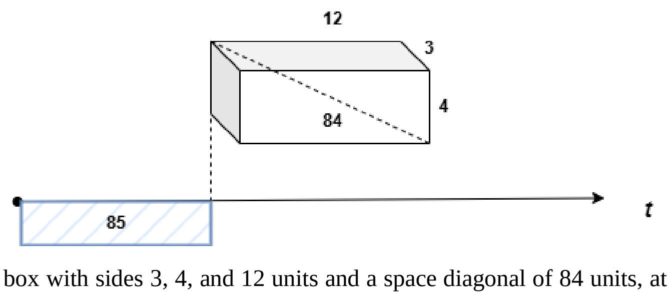

Fig. 1. A rectangular box with sides 3, 4, and 12 units and a space diagonal of 84 units, at point in time associated with the box's spatial configuration in the time unit 85, forming a Pythagorean quintuple.

This configuration highlights a practical, four-dimensional interpretation of the Pythagorean quintuple equation $a^2 + b^2 + c^2 + d^2 = e^2$. As shown in Fig. 1, the diagonal across the box (also known as the space diagonal) corresponds to the hypotenuse in three dimensions, derived by applying the Pythagorean theorem successively in two orthogonal planes. This metaphor illustrates the structure of a Pythagorean quintuple as an evolving spatial figure with a temporal component.

In this research a unique method is shown for generating Pythagorean Quintuples with only one variable. The method is based on the Platonic sequence ([2]) and using the basic formulas accordingly, with some algebraic development. The method does not cover all primitive cases, but the uniqueness of it is of using one variable, and more importantly, its simplicity.

Building on previous work ([3]) that introduced a novel single-variable method for generating quadruples using extensions of the Platonic sequence, this research presents a new set of formulas that continue this recursive construction to generate valid Pythagorean quintuples. Like the method used for quadruples, these formulas are based on a distinction between even and odd input values and further divide into cases based on the parity of intermediary components. The method's strength lies in its simplicity: a single input variable x yields all five components of the quintuple through closed-form algebraic expressions.

A Python implementation was developed as part of this study to systematically generate and verify these quintuples. Tables in the Results Section demonstrate the output of this implementation for a range of input values. Full software and data are available in ([4]).

2 Background and Related Work

Pythagorean quintuples, defined as integer solutions to the equation $a^2 + b^2 + c^2 + d^2 = e^2$, have been explored through various mathematical frameworks. One approach involves the use of quaternions to represent and generate these quintuples. [5] discusses the application of quaternion algebra in constructing Pythagorean quintuples, building upon the complex number representations used for Pythagorean triples.

Another significant contribution is the polynomial parametrization of Pythagorean quintuples. [6] provides a method for parametrizing Pythagorean quintuples using integer-valued polynomials, extending the classical results known for triples and quadruples. Additionally, empirical research by [7] demonstrates that simple algebraic identities can be utilized to self-generate Pythagorean n-tuples, including quintuples. This approach highlights the potential of basic algebraic expansions in generating these tuples without relying on complex formulas.

Furthermore, [8] presents a direct method to generate Pythagorean triples, which they generalize to Pythagorean quadruples and n-tuples. Their technique leverages the fact that the difference between the lengths of the hypotenuse and one leg of a Pythagorean triangle can have distinct values depending on the length of the other leg, a property that extends to higher-dimensional tuples.

These studies collectively enhance the understanding of Pythagorean quintuples, offering various methodologies for their generation and analysis.

[9] derives an exact formula for counting the number of distinct primitive Pythagorean quintuples, i.e., integer solutions $(\mathrm{w, x, y, z, t})$ to the equation $w^2 + x^2 + y^2 + z^2 = t^2$, under the condition that the greatest common divisor of all components is 1. Building on earlier results for triples and quadruples. In this paper a sophisticated counting mechanism is developed using Möbius inversion, quadratic forms, and twisted Euler functions with Dirichlet characters. The paper presents detailed derivations, provides exact and asymptotic formulas, and includes numerical tables illustrating the cumulative counts of primitive quintuples for values up to $t < 1000$.

In current mathematical research, Pythagorean tuples are explored not only for their theoretical significance but also for their practical applications in computer science, especially in areas such as optimization and cryptography, where lattice-based structures play a key role [10]. Recent advancements have introduced algorithms that efficiently generate and enumerate Pythagorean n-tuples through recursive methods and matrix-based transformations [11]. These techniques are closely connected to the broader study of expressing integers as sums of squares, a topic deeply rooted in analytic number theory [12]. Furthermore, research has explored generalizations to Pythagorean n -tuples, where the sum of $n - 1$ squares equal the square of the n^{th} term, expanding the scope of study and enabling applications in higher-dimensional geometry and theoretical physics [13].

A recent study by [14] proposed a matrix-based method for generating generalized Pythagorean tuples, including quintuples. This approach employs linear algebraic transformations combined with recursive iteration to efficiently compute valid tuples within specified bounds. The research further analyzes computational complexity and introduces code-level optimizations tailored for large-scale enumeration tasks. In parallel, implementations of such generators have appeared in platforms like Python, Mathematica, and MATLAB, supporting both educational and recreational exploration of these number sets through visualization and simulation tools [15]. These software-based tools serve pedagogical and research functions alike, often highlighting the interplay between number theory and geometry in three-dimensional space. Additionally, the generation of Pythagorean quintuples has been explored in the context of cryptographic applications—particularly in the design of hard lattice problems and obfuscation schemes, though this area remains in the early stages of development [16].

3 A formula for generating Pythagorean Quintuples

3.1 The Platonic sequence

The formula for generating Pythagorean triplets (The Platonic sequence) is inherently straightforward. Pythagoras himself developed the formula for odd lengths of the a side (perpendicular) around 540 BC, while Plato later introduced the formula for even a side length around 380 BC. The fundamental premise is that for any given integer a with a length of at least 3, a Pythagorean triplet can be derived using this formula for odd 'a' side length:

$$b = \frac{a^2-1}{2}\ ; c = b + 1 \tag{1}$$

And for the case of even 'a' side length, the formula is:

$$b = \left(\frac{a}{2}\right)^2 - 1\ ; c = b + 2 \tag{2}$$

As can be seen, these formulas can be manufactured by using on one chosen variable, that is a with a length of at least 3.

On the premise of these basic formulas, a bit more complex formulas were developed for generating Pythagorean Quadruples in [3].

3.2 Pythagorean Quadruples

Pythagorean quadruples, defined as integer solutions (a, b, c, d) to the Diophantine equation $a^2+b^2+c^2=d^2$ represent a natural generalization of the classical Pythagorean triples.

In [3] the formulas, that are a development of the Platonic sequence, are for the first variable, that is a with a length of at least 3.

For the base ground of the sequence, two options exist for a: a is odd, or a is even.

a is odd:

The b side is calculated as regularly in the Platonic sequence for odd a, and c, d by [3]:

$$b = \frac{a^2-1}{2}\ \ ; c = \frac{a^4}{8} + \frac{a^2}{4} - \frac{3}{8}\ ; d = c + 1 = \frac{a^4}{8} + \frac{a^2}{4} + \frac{5}{8} \tag{3}$$

a is even:

The b side is calculated as regularly in the Platonic sequence for even a is $b = \left(\frac{a}{2}\right)^2 - 1$. At this point, two options exist by [3] for the c side and the d side, as the b side can be either odd or even.

***a* is even, *b* is even:**

$$c = \frac{a^4}{64} + \frac{a^2}{8} - \frac{3}{4} \quad ; d = c + 2 = \frac{a^4}{64} + \frac{a^2}{8} + 1\frac{1}{4} \tag{4}$$

***a* is even, *b* is odd:**

$$c = \frac{a^4}{32} + \frac{a^2}{4} \quad ; d = c + 1 = \frac{a^4}{32} + \frac{a^2}{4} + 1 \tag{5}$$

3.3 Pythagorean Quintuples

Pythagorean quintuples, defined as integer solutions (a, b, c, d, e) to the Diophantine equation $a^2 + b^2 + c^2 + d^2 = e^2$ represent a natural generalization of the classical Pythagorean triples and their higher-dimensional analogue, the Pythagorean quadruples.

In [3] for the formulas developed for generating quadruples are derived from a recursive extension of the Platonic sequence. These formulas are constructed around a single input variable—denoted a —with a minimum length of 3.

As with the previous work on quadruples, the construction of quintuples also begins with a distinction between two fundamental cases: a is odd, or a is even (and then two fundamental cases of b being odd or even). Each case leads to a unique progression of formulas, which recursively build the remaining components of the quintuple through algebraic expressions that preserve the structure of the original Diophantine identity.

For simplifying, let *x* be the variable for which the algebraic calculation is performed. *x* is with a length of at least 3. For all formulas the first side a is set as:

$$\boldsymbol{a = x} \tag{6}$$

At this point, two options exist for *x* (and a): *x* is odd, or *x* is even.

<u>*x* is odd</u>:

The b side is calculated as regularly in the Platonic sequence for odd a, and the c side is calculated by [3]:

$$\boldsymbol{b = \frac{x^2-1}{2}} \quad ; \quad \boldsymbol{c = \frac{x^4}{8} + \frac{x^2}{4} - \frac{3}{8}} \tag{7}$$

To continue a viable Platonic sequence for a quintuple, the d side from the Pythagorean quadruples, that is $d = c + 1 \rightarrow d = \frac{x^4}{8} + \frac{x^2}{4} + \frac{5}{8}$ is now used as the free variable ($a = x$) on the b side formula of the triples:

$$\boldsymbol{d} = \frac{\left(\frac{x^4}{8} + \frac{x^2}{4} + \frac{5}{8}\right)^2 - 1}{2} = \frac{x^8 + 4x^6 + 14x^4 + 20x^2 - 39}{128} \tag{8}$$

And the last formula for the case of *x* is odd, for the e side, as d side was the equivalent of b side formula of the triples, the e side is the equivalent of c side formula of the triples:

$$\boldsymbol{e} = d + 1 = \frac{x^8 + 4x^6 + 14x^4 + 20x^2 + 89}{128} \tag{9}$$

<u>*x* is even</u>:

The b side is calculated as regularly in the Platonic sequence for even a:

$$\boldsymbol{b = \left(\frac{x}{2}\right)^2 - 1} \; ; \tag{10}$$

At this point, two options exist for the c side, as the b side can be either odd or even.

***x* is even, *b* is even:**

The c side is calculated by [3]:

$$\boldsymbol{c} = \frac{x^4}{64} + \frac{x^2}{8} - \frac{3}{4} \tag{11}$$

To continue a viable Platonic sequence for a quintuple, the d side from the Pythagorean quadruples, that is $d = c + 2 \ \rightarrow d = \frac{x^4}{64} + \frac{x^2}{8} + 1\frac{1}{4}$ is now used as the free variable ($a = x$) on the b side formula of the triples:

$$\boldsymbol{d} = \left(\frac{\frac{x^4}{64}+\frac{x^2}{8}+1\frac{1}{4}}{2}\right)^2 - 1 = \frac{x^8}{16384} + \frac{x^6}{1024} + \frac{7x^4}{512} + \frac{5x^2-39}{64} \tag{12}$$

And the last formula for the case of *x* is even and b is even for the *e* side, as *d* side was the equivalent of *b* side formula of the triples, the *e* side is the equivalent of *c* side formula of the triples:

$$\boldsymbol{e} = d + 2 = \frac{x^8}{16384} + \frac{x^6}{1024} + \frac{7x^4}{512} + \frac{5x^2+89}{64} \tag{13}$$

***x* is even, *b* is odd:**

The c side is calculated by [3]:

$$\boldsymbol{c} = \frac{x^4}{32} + \frac{x^2}{4} \tag{14}$$

To continue a viable Platonic sequence for a quintuple, the d side from the Pythagorean quadruples, that is $d = c + 1 \ \rightarrow d = \frac{x^4}{32} + \frac{x^2}{4} + 1$ is now used as the free variable ($a = x$) on the b side formula of the triples:

$$\boldsymbol{d} = \frac{(\frac{x^4}{32}+\frac{x^2}{4}+1)^2-1}{2} = \frac{x^8}{2048} + \frac{x^6}{128} + \frac{x^4}{16} + \frac{x^2}{4} \tag{15}$$

And the last formula for the case of x is odd, for the e side, as d side was the equivalent of b side formula of the triples, the e side is the equivalent of c side formula of the triples:

$$\boldsymbol{e} = d + 1 = \frac{x^8}{2048} + \frac{x^6}{128} + \frac{x^4}{16} + \frac{x^2}{4} + \mathbf{1} \tag{16}$$

3.4 Proof

In all the cases presented above the equations need to be checked in terms of adherence to the Pythagorean Quadruple equation of $a^2 + b^2 + c^2 + d^2 = e^2$.

***x* is odd:**

$$\boldsymbol{a^2} + \boldsymbol{b^2} + \boldsymbol{c^2} + \boldsymbol{d^2} = x^2 + \left(\frac{x^2-1}{2}\right)^2 + \left(\frac{x^4}{8} + \frac{x^2}{4} - \frac{3}{8}\right)^2 + \left(\frac{x^8+4x^6+14x^4+20x^2-39}{128}\right)^2 =$$
$$\frac{x^{16}+7921}{16384} + \frac{267x^8}{8192} + \frac{11x^{12}+723x^4}{4096} + \frac{x^{14}+19x^{10}+159x^6+445x^2}{2048} \tag{17}$$

$$\boldsymbol{e^2} = \left(\frac{x^8+4x^6+14x^4+20x^2+89}{128}\right)^2 =$$

$$\frac{x^{16}+7921}{16384} + \frac{267x^8}{8192} + \frac{11x^{12}+723x^4}{4096} + \frac{x^{14}+19x^{10}+159x^6+445x^2}{2048} \tag{18}$$

***x* is even, *b* is even:**

$$\boldsymbol{a^2+b^2+c^2+\ d^2} = x^2 + \left(\left(\frac{x}{2}\right)^2 - 1\right)^2 + \left(\frac{x^4}{64} + \frac{x^2}{8} - \frac{3}{4}\right)^2 + \left(\frac{x^8}{16384} + \frac{x^6}{1024} + \frac{7x^4}{512} + \frac{5x^2-39}{64}\right)^2 = \boldsymbol{\frac{x^{16}}{268435456} + \frac{x^{14}}{8388608} + \frac{11x^{12}}{4194304} + \frac{19x^{10}+267x^8}{524288} + \frac{159x^6}{32768} + \frac{723x^4}{16384} + \frac{445x^2}{2048} + \frac{7921}{4096}} \quad (19)$$

$$\boldsymbol{e^2} = \left(\frac{x^8}{16384} + \frac{x^6}{1024} + \frac{7x^4}{512} + \frac{5x^2+89}{64}\right)^2 = \boldsymbol{\frac{x^{16}}{268435456} + \frac{x^{14}}{8388608} + \frac{11x^{12}}{4194304} + \frac{19x^{10}+267x^8}{524288} + \frac{159x^6}{32768} + \frac{723x^4}{16384} + \frac{445x^2}{2048} + \frac{7921}{4096}} \quad (20)$$

***x* is even, *b* is odd:**

$$\boldsymbol{a^2+b^2+c^2+\ d^2} = x^2 + \left(\left(\frac{x}{2}\right)^2 - 1\right)^2 + \left(\frac{x^4}{32} + \frac{x^2}{4}\right)^2 + \left(\frac{x^8}{2048} + \frac{x^6}{128} + \frac{x^4}{16} + \frac{x^2}{4}\right)^2 = \boldsymbol{\frac{x^{16}}{4194304} + \frac{x^{14}}{131072} + \frac{x^{12}}{8192} + \frac{5x^{10}}{4096} + \frac{9x^8}{1024} + \frac{3x^6}{64} + \frac{3x^4}{16} + \frac{x^2}{2} + 1} \quad (21)$$

$$\boldsymbol{e^2} = \left(\frac{x^8}{2048} + \frac{x^6}{128} + \frac{x^4}{16} + \frac{x^2}{4} + 1\right)^2 = \boldsymbol{\frac{x^{16}}{4194304} + \frac{x^{14}}{131072} + \frac{x^{12}}{8192} + \frac{5x^{10}}{4096} + \frac{9x^8}{1024} + \frac{3x^6}{64} + \frac{3x^4}{16} + \frac{x^2}{2} + 1} \quad (22)$$

Table 1. Pythagorean Quintuples in the range of a < 100.

a	*b*	*c*	*d*	*e*
59	1740	1515540	1148432261340	1148432261341
95	4512	10183584	51852701726112	51852701726113
4	3	12	84	85
19	180	16380	134168580	134168581
70	1224	375768	35300773224	35300773226
48	575	166464	13855298112	13855298113
91	4140	8573940	36756232135740	36756232135741
65	2112	2232384	2491771394112	2491771394113
73	2664	3551112	6305201769384	6305201769385
69	2380	2834580	4017424722780	4017424722781
21	220	24420	298192620	298192621
35	612	187884	17650386612	17650386613
84	1763	1557612	1213079128884	1213079128885

Table 2. Pythagorean Quintuples in the range of 100 < a < 200.

a	***b***	***c***	***d***	***e***
182	8280	17147880	73512464271480	73512464271482
106	2808	1974024	974194662168	974194662170
163	13284	88245612	3893644106872880	3893644106872881
139	9660	46667460	1088925958093260	1088925958093261
159	12640	79897440	3191800539174240	3191800539174241
128	4095	8392704	35218748608512	35218748608513
130	4224	4464768	4983542788224	4983542788226
137	9384	44039112	969721736913384	969721736913385
105	5512	15196584	115468097831112	115468097831113
199	19800	196039800	19215801788059800	19215801788059801
138	4760	5669160	8034849445560	8034849445562
192	9215	42476544	902128437568512	902128437568513
166	6888	11868024	35212510284168	35212510284170

4 Results

As part of this research, a Python-based program was developed to implement the various formulaic constructions presented in this work. The software covers all identified variants, and additional implementation details are provided in [4], along with a supplementary dataset containing extended results. Illustrative examples of the program's output—demonstrating the behavior of the formulas across different input values—are summarized in Table 1 for Pythagorean Quintuples in the range of a < 100, Table 2 for Pythagorean Quintuples in the range of 100 < a < 200, and in Table 3 for Pythagorean Quintuples in the range of 200 < a < 300.

A visual representation of a quintuple is provided in Fig. 1, where the geometric interpretation of the four spatial dimensions is extended into time. Here, e can be interpreted as the point in time associated with the spatial configuration represented by the first four components. This temporal extension provides a dynamic lens through which to conceptualize the transition from quadruples to quintuples.

The results demonstrate the robustness of the method across a wide numerical domain and suggest that the proposed formulas may serve as a basis for generating quintuples in further computational and theoretical investigations.

Table 3. Pythagorean Quintuples in the range of 200 < a < 300.

a	*b*	*c*	*d*	*e*
225	25312	320373984	51319745132390100	51319745132390101
237	28084	394383612	77769217101466800	77769217101466801
203	20604	212283012	22532038804179000	22532038804179001
243	29524	435862812	94988195878136400	94988195878136401
244	14883	110781612	6136282889440880	6136282889440881
259	33540	562499340	158202754312717000	158202754312717001
298	22200	123232200	3796543902442200	3796543902442202
208	10815	58503744	1711344089512510	1711344089512511
292	21315	227207244	25811566090245000	25811566090245001
297	44104	972625512	473000194269256000	473000194269256001
256	16383	134234112	9009398546448380	9009398546448381
273	37264	694340112	241054096260426000	241054096260426001
247	30504	465277512	108241582051732000	108241582051732001

5 Conclusion and Future Work

In the initial stages of this research, a foundational scheme plan was proposed in [17], with several ideas from the scheme further developed in [18] and [19]. The basis for this research was done in [19], where the general algorithmic heuristic was established along with several initial results.

This paper introduced a novel and accessible method for generating Pythagorean quintuples using a single-variable approach inspired by the classical Platonic sequence. By extending the structure of formulas traditionally used to produce Pythagorean triples—and later quadruples—we developed compact algebraic expressions capable of generating quintuples across both even and odd input cases. While the method does not account for all primitive solutions, it offers a compelling balance between algebraic elegance and computational feasibility.

A key strength of the approach lies in its simplicity: valid quintuples are derived through direct algebraic manipulation of a single input variable, making the method highly suitable for both educational contexts and algorithmic exploration. To demonstrate its effectiveness, we developed a Python-based tool that generates a broad set of valid quintuples and verifies the results through symbolic computation.

For future work, a relatively ambitious goal is to define a general construction or parametrization for Pythagorean n-tuples, where $n-1$ squared terms sum to the square of the n^{th} term: ${a_1}^2 + {a_2}^2 + \cdots + {a_{n-1}}^2 = {a_n}^2$. As n increases, the complexity of the problem escalates significantly—both in algebraic structure and computational demands. Advancing this line of research may require the application of matrix transformations, higher-dimensional lattice frameworks, or recursive algorithmic techniques to uncover generalized generation methods. An additional promising avenue lies in the use of symbolic computation and machine learning to detect patterns in existing solutions, potentially giving rise to new conjectures or partial parametrizations.

Developing such generalized formulas would not only deepen our understanding of Diophantine equations in higher dimensions but also expand the practical relevance of these mathematical structures in fields such as cryptography, error-correcting codes, and theoretical physics.

A promising avenue for future exploration involves investigating diverse communication systems requiring encryption solutions, particularly those employing complex and non-trivial communication protocols, as highlighted in [20].

References

1. Mordell, L. J. (1969). *Diophantine Equations: Diophantine Equations* (Vol. 30). Academic press.
2. Barnett, J. H. (2017). Generating Pythagorean triples: the methods of Pythagoras and of Plato via Gnomons.
3. Voloch, N., (2025) Generating Pythagorean Quadruples with One Variable. arXiv preprint. https://arxiv.org/submit/6311235/preview
4. https://github.com/nadavvoloch/PythagoreanQuintuples
5. Sporn, M. (2015). Quaternions and Pythagorean quintuples. *International Journal of Pure and Applied Mathematics*, 101(4), 555-563.
6. Frisch, S., & Vaserstein, L. (2011). Polynomial parametrization of Pythagorean quadruples, quintuples, and sextuples. *arXiv preprint arXiv:1106.3493*.
7. Wadhawan, N. (2023). Self-generating n-tuples. Graduate Journal of Interdisciplinary Research, Reports and Reviews, 1(1), 18-27.
8. Roy, T., & Sonia, F. J. (2012). A direct method to generate Pythagorean triples and its generalization to Pythagorean quadruples and n-tuples. *arXiv preprint arXiv:1201.2145*.
9. Hürlimann, W. E. R. N. E. R. (2015). On the number of primitive Pythagorean quintuples. *Journal of Algebra, Number Theory: Advances and Applications*, *13*(1), 13-28.
10. Nguyen, P. Q., & Vallée, B. (Eds.). (2010). The LLL Algorithm: Survey and Applications. Springer.
11. Weisstein, E. W. (n.d.). Pythagorean n-Tuple. MathWorld. Retrieved from https://mathworld.wolfram.com/PythagoreanNTuple.html
12. Hardy, G. H., & Wright, E. M. (2008). An Introduction to the Theory of Numbers (6th ed.). Oxford University Press.
13. Dickson, L. E. (2005). History of the Theory of Numbers, Vol. II: Diophantine Analysis. Dover Publications.
14. Ghosh, M., & Pal, S. (2023). Generalized Pythagorean tuples via matrix transformations. *arXiv preprint arXiv:2303.05776*. https://arxiv.org/abs/2303.05776
15. Elkin, M. (2018). Integer lattice visualization for Pythagorean tuples. *Mathematics in Computer Science*, *12*(2), 235–248. https://doi.org/10.1007/s11786-017-0345-5
16. Nguyen, P. Q., & Regev, O. (2010). Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *Journal of Cryptology*, *23*(1), 1–25. https://doi.org/10.1007/s00145-009-9048-5
17. Voloch, Benjamin (1996), Ben Gurion University of the Negev, & Baruch, Moshe (2005), The Hebrew University of Jerusalem, Israel. Private communication.
18. Voloch, N., Birnbaum, E., & Sapir, A. (2014, December). Generating error-correcting codes based on tower of Hanoi configuration graphs. In 2014 IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI) (pp. 1-4). IEEE.
19. Voloch, N., & Raicol, N. V. B. (2024, September). A Cryptographic Encryption Scheme based on a Pythagorean Triplets Manufacturing Formula. In *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)* (pp. 529-534). IEEE.
20. Voloch, N., & Hajaj, M. M. (2022, November). Handling Exit Node Vulnerability in Onion Routing with a Zero-Knowledge Proof. In International Conference on Information Integration and Web (pp. 399-405). Cham: Springer Nature Switzerland.