

Tabulating $x = c - a$ helps with computing later Pythagorean triples from earlier ones in a given $c - b$ class

John van der Linden

October 14, 2026

Consider the k -indexed primitive Pythagorean triples $a_k^2 + b_k^2 = c_k^2$ where $c_k > b_k > a_k$ and $y = 1$ such that $c_k = b_k + y = b_k + 1$. Start with the first triple $3^2 + 4^2 = 5^2$. Here, $a_k = a_1 = 3$, $b_k = b_1 = 4$, and $c_k = c_1 = 5$. Now define the positive integer value x_k such that $c_k = x_k + a_k = y + b_k$. For $k = 1$, $x_1 = c_1 - a_1 = 5 - 3 = 2$. We can write these values as the first row in a table, preceded by the index k and followed by the parameter y .

k	a_k	b_k	c_k	x_k	y

1	3	4	5	2	1

In the process of experimenting with $y = 1$ primitive triples tabulated successively in this way (with the triple in row $k + 1$ being the immediate sequential successor to the one in row k), I found that the following formulas could be used to find x_p and c_p , where (in this case) $p = k + 1$:

- (1) $x_p = a_k + c_k$
- (2) $c_p = (x_p^2 + y^2) / c_k$

For example, from the first row in our table (where $k = 1$), x_p and c_p compute as follows:

- (3) $x_2 = a_1 + c_1 = 3 + 5 = 8$
- (4) $c_2 = (x_2^2 + y^2) / c_1 = (8^2 + 1^2) / 5 = 65 / 5 = 13$

Now we can easily find a_2 and b_2 (since $c_k = x_k + a_k = y + b_k$):

- (5) $a_2 = c_2 - x_2 = 13 - 8 = 5$
- (6) $b_2 = c_2 - y = 13 - 1 = 12$

k	a_k	b_k	c_k	x_k	y

1	3	4	5	2	1
2	5	12	13	8	1

Next, going from $k = 2$ to $k = 3$:

- (7) $x_3 = a_2 + c_2 = 5 + 13 = 18$
- (8) $c_3 = (x_3^2 + y^2) / c_2 = (18^2 + 1^2) / 13 = 325 / 13 = 25$
- (9) $a_3 = c_3 - x_3 = 25 - 18 = 7$
- (10) $b_3 = c_3 - y = c_3 - 1 = 25 - 1 = 24$

k	a_k	b_k	c_k	x_k	y

1	3	4	5	2	1
2	5	12	13	8	1
3	7	24	25	18	1

At least up to $k = 9$ (as far as I tested it), this method appears to generate every successive primitive triple in the $y = 1$ group without skipping any of them. What about the $y = 2$ group? Start with $8^2 + 15^2 = 17^2$.

k	a_k	b_k	c_k	x_k	y
1	8	15	17	9	2

Apply steps 3-6 from above, except with $y = 2$:

$$\begin{aligned}
 (11) \quad x_2 &= a_1 + c_1 = 8 + 17 = 25 \\
 (12) \quad c_2 &= (x_2^2 + y^2) / c_1 = (25^2 + 2^2) / 17 = 629 / 17 = 37 \\
 (13) \quad a_2 &= c_2 - x_2 = 37 - 25 = 12 \\
 (14) \quad b_2 &= c_2 - y = 37 - 2 = 35
 \end{aligned}$$

k	a_k	b_k	c_k	x_k	y
1	8	15	17	9	2
2	12	35	37	25	2

Repeat with the new $k = 2$ row to generate the $k = 3$ row:

$$\begin{aligned}
 (15) \quad x_3 &= a_2 + c_2 = 12 + 37 = 49 \\
 (16) \quad c_3 &= (x_3^2 + y^2) / c_2 = (49^2 + 2^2) / 37 = 2405 / 37 = 65 \\
 (17) \quad a_3 &= c_3 - x_3 = 65 - 49 = 16 \\
 (18) \quad b_3 &= c_3 - y = 65 - 2 = 63
 \end{aligned}$$

k	a_k	b_k	c_k	x_k	y
1	8	15	17	9	2
2	12	35	37	25	2
3	16	63	65	49	2

After further successful trials with $y = 1$ and $y = 2$, I wondered if (1) and (2) would generate all primitive triples for any value of y having primitive solutions. Investigating $y > 2$, I found that the formulas do work, but $p > k + 1$ for the computed values x_p and c_p . For example, when $y = 8$, $p = k + 2$, so that applying (1) and (2) from $k = 1$ generates the $k = 3$ row; from $k = 3$ one gets the $k = 5$ row; and so on. However, if one starts additional computation sequences with (1) and (2) at each row skipped by the $k = 1$ step, my tabulations suggested that all primitive triples *may* be generated. If this is true, then all that is needed is a sufficient number (call it S_y , where $p = k + S_y$) of initial seed triples.

Is there a way to predict the value of S_y ? Below, I list provisional values of S_y ($1 \leq y \leq 50$) grouped by y -category (odd perfect square vs. double perfect square) and including the root r of the perfect square. I tentatively determined S_y for each y by gathering lists of the first several triples (calculated from Euclid's formula); counting how many of these triples were skipped when applying (1) and (2) in the initial $k = 1$ step; and then starting additional computation sequences with some or all of the skipped triples to confirm that they generated later triples at the expected values of k .

Odd perfect squares y			Double perfect squares y			Thus it appears that, at least when $1 \leq y \leq 50$, the tentative values of S_y can be predicted from r as follows:
y	r	S_y	y	r	S_y	
-----			-----			
1	1	1	2	1	1	(19) $r = 1:$ $S_y = r$ (20) $r > 1$ and even: $S_y = r$ (21) $r > 1$ and odd: $S_y = r - 1$
9	3	2	8	2	2	
25	5	4	18	3	2	
49	7	6	32	4	4	
			50	5	4	

For $1 \leq y < 350$, here are tentatively determined ($1 \leq y \leq 50$) and predicted ($50 < y < 350$) values of S_y , sorted by r ascending. I did not make predictions for $y > 350$, but have still included placeholder rows for these. In a partial test of $y = 289$, using a list (prepared with Euclid's formula) of the first 17 triples for this y-value, I found that applying (1) and (2) to the $k = 1$ triple generated the $k = 17$ triple, suggesting that $S_{289} = 16$, which matches the predicted value.

r	y	S_y	S_y category

1	1	1	determined
1	2	1	determined
2	8	2	determined
3	9	2	determined
3	18	2	determined
4	32	4	determined
5	25	4	determined
5	50	4	determined
6	72	6	predicted
7	49	6	determined
7	98	6	predicted
8	128	8	predicted
9	81	8	predicted
9	162	8	predicted
10	200	10	predicted
11	121	10	predicted
11	242	10	predicted
12	288	12	predicted
13	169	12	predicted
13	338	12	predicted
14	392	--	N/A
15	225	14	predicted
15	450	--	N/A
16	512	--	N/A
17	289	16	predicted

Discussion

It seems likely that (1) and (2) are simply reformulations of (or can be derived from) other methods for computing primitive triples, such as Euclid's formula or the Berggren/Barning matrix transformations. I haven't attempted such a derivation, and in my cursory research into the various known triple-generating methods, I didn't find this method clearly discussed, so I describe it here. If it has been previously described elsewhere, please let me know!

When investigating this system, I didn't go very deeply into the set of triples for any given value of y , and of course it was impossible to check them all, since there are infinitely many. Thus I cannot state with certainty that the formulas (1) and (2) will generate *all* the primitive triples for a given y , using only the initial S_y seed triples.

This method has the advantage that, for a given value of y , once the initial set of S_y seed triples is assembled (e.g. with Euclid's formula), many subsequent triples can be computed solely from the seed triples. There are also only two main formulas to remember ((1) and (2)), and they are fairly straightforward.

I would expect this method to be especially useful for $y \leq 18$, where only 1 or 2 seed triples (and thus only 1 or 2 independent computation sequences) are needed.

References and Acknowledgments

Google's "AI Overview" tool was used to generate or validate certain tables of primitive triples (not including x -values) and learn about previously described triplet-generating formulas (including the Euclid and Berggren/Barning processes). It also taught me the "odd perfect squares" and "double perfect squares" categories for the difference $c - b$. Wikipedia (https://en.wikipedia.org/wiki/Pythagorean_triple and https://en.wikipedia.org/wiki/Tree_of_primitive_Pythagorean_triples) helped me improve my knowledge of Pythagorean triples and learn about additional triplet-generating methods after I developed (1) and (2).