

Graeco-Latin Square Poems

Lisa Lajeunesse

School of STEM, Capilano University, North Vancouver, Canada; llajeune@capilanou.ca

Abstract

This paper introduces a new poetic form based on a Graeco-Latin square. I show how to read the poem from the square and how such a poem can be encoded into a poetry puzzle. I then illustrate how to reconstruct the Graeco-Latin square from the poem's stanza form. Finally, I share some techniques that have helped me to create Graeco-Latin square poems.

Introduction

At Bridges 2018, I introduced the idea of *poetry puzzles* [1]. These are logic puzzles, similar to *Sudoku*, that are based on combinatorial designs. The solved puzzle unscrambles the words of a poem. One reviewer's comment brought Brian Isett's *Grid Poems* to my attention. This inspired me to create another type of poetry puzzle based on a Graeco-Latin square. This new poetic form was included in the workshop at Bridges Stockholm though not in the proceedings paper. The form and its poetry puzzle variant are presented here.

Each of Isett's grid poems are generated from an $n \times n$ square of *cells* in which each cell contains a line of the poem. The poem is then generated by reading the lines twice from the square, once from left to right, and again from top to bottom. A sample grid poem by Isett can be found on his webpage [2]. This paper describes a square grid in which the cells are read twice as well. But for these poems, the order in which the cells are read is determined by a Graeco-Latin square. Such poems can also be encoded as poetry puzzles. In the examples presented here, the cells contain one word (the occasional cell contains two words) of the poem rather than full lines as in Isett's grid poems, although this is not a requirement of the Graeco-Latin square form.

Because of their shape, Graeco-Latin square poems fall within a category known as *pattern poetry*. Sarah Glaz's survey article *Mathematical Pattern Poetry* [3] traces the long tradition of pattern poetry, beginning with *The Sator Square* dating from 79 AD, through to the present day. One contemporary example is the poem *Square Math Problem* by JoAnne Growney. Like Isett's grid poems and Graeco-Latin square poems, both *The Sator Square* and Growney's poem are based on the geometry of a square. The square can be implied, as in Growney's poem, rather than explicitly given as in *The Sator Square*. The cells in the square of *The Sator Square* contain individual letters that together form a palindrome in Latin. In Growney's poem they contain individual syllables. In the Graeco-Latin square poems they contain a single word and in Isett's grid poems they may contain entire lines of poetry.

Each $n \times n$ square produces stanza(s) consisting of n lines of poetry, where each line contains either n letters, n syllables, n words or n lines. These poems can be distinguished by the terms *letter-square*, *syllable-square*, *word-square* and finally, *line-square*. The Graeco-Latin square poems given here are comprised of cells containing one or two words, so that each stanza consists of n lines containing approximately n words per line. Therefore, they are *word-square* poems.

Graeco-Latin Squares

We begin with an $n \times n$ grid of cells in which each cell contains a number and a letter. The number is chosen from the set $S = \{1, 2, \dots, n\}$ and the letter is chosen from the set $T = \{A, B, \dots, \Delta\}$ where Δ is the n^{th} letter (note that in this paper, $n = 3$ or 4). We can regard the number-letter combination in each cell as

an ordered pair from the Cartesian product $S \times T$. The $n \times n$ square is a Graeco-Latin square if it satisfies the following three conditions:

- (1) Each number from S appears exactly once in each row and column of the grid.
- (2) Each letter from T appears exactly once in each row and column of the grid.
- (3) Each number-letter combination (note that there are n^2 such pairs) appears exactly once in the square.

Condition (1) ensures that the numbers form a Latin square of order n on the symbols $S = \{1, 2, \dots, n\}$ and (2) ensures that the letters form a Latin square of order n on the symbols $T = \{A, B, \dots, \Delta\}$. If two Latin squares of order n satisfy condition (3) when combined into a single square then that square is Graeco-Latin and the separate Latin squares are referred to as a pair of *orthogonal* Latin squares. Figure 1 shows an example of a Graeco-Latin square of order $n = 3$.

2B	1C	3A
1A	3B	2C
3C	2A	1B

Figure 1: Graeco-Latin square of order 3.

Graeco-Latin squares have a rich history, and have been studied extensively since Leonard Euler first explored their connection to Magic squares in 1782 [4]. Euler conjectured that Graeco-Latin squares of order $n \equiv 2 \pmod{4}$ did not exist. His conjecture was proven false in 1959 when E.T. Parker, and then R.C. Bose and S.S. Shrikhande constructed Graeco-Latin squares of order 10 and order 22 respectively [5]. News of their work reached graph theorist Claude Berge in Paris. Berge was a member of the French literary group *Oulipo* and he showed Parker's order 10 Graeco-Latin square to his fellow group members [6]. A young Georges Perec was present and the prospect of using this square as a basis for a novel caught his imagination. The Graeco-Latin square made its literary debut in Perec's masterful novel *La Vie: Mode d'Emploi* [7]. Perec based a large part of the novel's structure on Parker's original square and the many Graeco-Latin squares that can be generated by permuting the rows and columns of the original [8].

The Graeco-Latin Square Poem

Figure 2 shows a Graeco-Latin square poem of order 4 represented in its *square form*. The numbers and letters form a Graeco-Latin square, but each cell also includes one or two words. To reconstruct the poem in *stanza form*, read the words from left to right based on the numbers inside the cells of the square to form the first stanza. To obtain the second stanza, the words are read a second time from top to bottom based on the letters inside the cells.

1D bruised	3C humanity	4A fragrant	2B waste
3B fuels	1A rare	2C not	4D earth
4C evolution's	2D —death	1B greenhouse	3A hour
2A expires	4B effect,	3D at midnight	1C flower

Figure 2: Graeco-Latin square poem.

For example, the words of the first line of the first stanza are paired in the above square with the number 1. When these words are read from left to right, (note that they lie along the main diagonal), we obtain the first line of the poem.

bruised rare greenhouse flower

In Figure 3, cells that contain the same number are given the same shading (with light to dark corresponding to numbers 1 through 4), and the arrow indicates the direction in which to read the words. The first stanza is

*bruised rare greenhouse flower
 expires —death not waste
 fuels humanity at midnight hour
 evolution's effect, fragrant earth*

1D bruised	3C humanity	4A fragrant	2B waste
3B fuels	1A rare	2C not	4D earth
4C evolution's	2D —death	1B greenhouse	3A hour
2A expires	4B effect,	3D at midnight	1C flower

Figure 3: Shading of cells by numbers.

The second stanza is reconstructed by reading the words paired with the appropriate letter from top to bottom. The words of the first line are paired with A, the second line with B and so on. Thus, the first line of the second stanza is

fragrant rare hour expires

In Figure 4, the cells are shaded once more from light to dark, this time according to their letters. The arrow reminds us to read from top to bottom.

*fragrant rare hour expires
 waste fuels greenhouse effect
 humanity not evolution's flower
 bruised earth —death at midnight*

1D bruised	3C humanity	4A fragrant	2B waste
3B fuels	1A rare	2C not	4D earth
4C evolution's	2D —death	1B greenhouse	3A hour
2A expires	4B effect,	3D at midnight	1C flower

Figure 4: Shading of cells by letters.

The complete poem in stanza form is

*bruised rare greenhouse flower
 expires —death not waste
 fuels humanity at midnight hour
 evolution's effect, fragrant earth*

*fragrant rare hour expires
 waste fuels greenhouse effect
 humanity not evolution's flower
 bruised earth —death at midnight*

Graeco-Latin Square Poems as Poetry Puzzles

Graeco-Latin square poems can be turned into poetry puzzles by presenting them in square form and including a minimal set of number-letter pairs and/or single symbols as clues. For the puzzle to be well-formed, there must be a unique way to fill in the missing symbols so that the ordered pairs form a Graeco-Latin square.

Figure 5 gives an example of such a puzzle of order $n = 3$. We must complete the Graeco-Latin square in order to unscramble the lines in the poem. As a first step, we use conditions (1) and (2) to fill in as many of the numbers and letters that we can. Condition (1) states that the numbers on their own form a Latin square. Condition (2) states that the letters also form a Latin square. Figure 6 shows the result of this first step which enables us to complete the numbers in column 2 (only a 2 can go in row 1 column 2, leaving a 1 in row 3 column 2) and the one missing letter in row 2 can be filled in (a C in column 1).

Now we are able to use the new clues, shown in Figure 6, to fill in all of the remaining numbers in the square. These new additions are shown in Figure 7.

1 Time will	Still	Swift
Hours	3A Never	B Slowly
End	Pass	Wait

Figure 5: Graeco-Latin square poetry puzzle.

1 Time will	② Still	Swift
Ⓒ Hours	3A Never	B Slowly
End	1 Pass	Wait

Figure 6: Partially-solved poetry puzzle.

1 Time will	2 Still	3 Swift
2C Hours	3A Never	1B Slowly
3 End	1 Pass	2 Wait

Figure 7: Partially-solved poetry puzzle.

We now tackle the missing letters. Consider the 2 in row 1, column 2. Figure 7 shows that it must be paired with either a B or C, since column 2 already contains the letter A. We realize that it must be paired with a B, since the combination 2C already appears once in the square (row 2 column 1), and each number-letter combination must appear exactly once by condition (3) of a Graeco-Latin square. With this information we can fill in the two missing letters in column 2; the additions are shown in Figure 8.

At this point, it is straightforward to fill in the remaining letters. The solved puzzle is shown in Figure 9.

1 Time will	2B Still	3 Swift
2C Hours	3A Never	1B Slowly
3 End	1C Pass	2 Wait

Figure 8: Partially-solved poetry puzzle.

From the solved puzzle, we can reconstruct the poem in stanza form.

*Time will pass slowly,
Hours still wait,
End never swift*

*Time will never wait
Still slowly end
Swift hours pass*

1A Time will	2B Still	3C Swift
2C Hours	3A Never	1B Slowly
3B End	1C Pass	2A Wait

Figure 9: Solved poetry puzzle.

Reconstructing the Graeco-Latin Square from Stanza Form

How can we identify whether or not a poem in stanza form is a Graeco-Latin square poem of order n ? There are some basic properties that any Graeco-Latin square poem must satisfy that can be easily checked to identify likely candidates. Firstly, the poem must consist of two stanzas, and each stanza must have the same number of lines n . Furthermore, if the poem has been generated from a square in which the cell entries are single words, then the number of words in each line must also equal n . Finally, the words of the second stanza must be a permutation of those in the first. These conditions are necessary but not sufficient for a poem to be a Graeco-Latin square poem. If a poem satisfies these conditions, then we look more closely to determine whether or not it is a Graeco-Latin square poem.

The next stage is to reverse the steps that we normally use to generate the stanza form of the poem. We illustrate this with the following poem. Note that each stanza has 3 lines and each line contains 3 words. Also, the words in the second stanza are a permutation of those in the first. Therefore, this could be a Graeco-Latin square poem in stanza form.

*Understand **reason** desire
Beyond infinite need
Measures heart's love*

*Understand heart's need
Infinite desire measures
Love beyond **reason***

The square form can be determined by examining each repeated word and its position in each stanza of the poem. We illustrate with the word *reason*; both occurrences are shown above in bold. In the 1st stanza it appears in the first line so it is paired with the number 1. In the second stanza it appears in the third line, so it is paired with the letter C. Thus the word *reason* appears in the same cell that contains the ordered pair 1C.

Now we determine which cell of the square contains this word and ordered pair. Because *reason* is the second word of its line in the 1st stanza, it must be located in the 2nd column of the square, since when we read from left to right, all of the second words of lines are read from the 2nd column. Similarly, because *reason* is the third word of its line in the 2nd stanza, it must appear in the 3rd row of the square, since when we read from top to bottom, all of the third words of lines are read from the 3rd row. Thus we conclude that the word *reason* and the ordered pair 1C appear in the 2nd column and 3rd row of the square.

A similar analysis for each of the repeated words of the poem enables us to reconstruct the square form of the poem shown in Figure 10, from which we can verify that the ordered pairs are arranged into a Graeco-Latin square. We can now confirm that the original poem is a Graeco-Latin square poem.

Note that the reconstruction assumes that each stanza is constructed from a set of n^2 distinct words.

By illustrating how the poem can be read from the Graeco-Latin square, and how the Graeco-Latin square can be reconstructed from the poem, we see that for a given placement of n^2 distinct words in a square, there is a one-to-one correspondence between how we can superimpose ordered number-letter combinations in the cells and the arrangement of the words into two stanzas.

1A Understand	2B Infinite	3C Love
2C Beyond	3A Heart's	1B Desire
3B Measures	1C Reason	2A Need

Figure 10: *Square form of poem.*

Writing a Graeco-Latin Square Poem

Creating a Graeco-Latin square poem is a challenge due to the stringent constraints of the form. I've shared a few hard won tricks and techniques below that have helped me to write the three poems in this paper. I've dubbed these techniques, in the order in which they are introduced, *Flexibility of vocabulary*, *Anchoring the poem*, *Word placement*, *Incremental improvements*, *Flexibility of the square* and finally, *Sleight of hand*.

Before writing, choose a Graeco-Latin square to work with. In other words, begin with a placement of the number-letter combinations into a square grid so that the arrangement of symbols forms a Graeco-Latin square. It's best to assume that the choice of square does not matter. While some squares will enable some poems and not others, it's difficult to predict which particular square offers the best poetic options. However, there is one observation to make about the choice of square. If the upper-left cell contains the 1A combination, each stanza of the poem will begin with the same word. This may or may not be desirable depending on the particular poem and the poet's aesthetic sensibility.

Flexibility of vocabulary: Once a square is fixed, choose appropriate vocabulary for the poem's topic or theme. Words with multiple meanings have more flexibility; this is desirable, since each word appears twice in the poem and multiple meanings will permit a different usage in each stanza. Keep in mind that you are not committed to using this vocabulary in the final version. Initial words may be swapped out of the poem at a later stage.

Anchoring the poem: Anchor the poem from the beginning with one or two good lines. Note that if we choose one line from each stanza of a finished poem, the form necessitates that the two lines will have exactly one word in common. For example, in the poem from the last section, the two lines *Beyond infinite need* and *Infinite desire measures* share the word *infinite*. If we want these two lines in a poem, we can begin by positioning the 5 words in the square to ensure these lines appear, one in each stanza.

Word placement: The fact that each line is read either from left to right or top to bottom, influences the placement of word types. For example, prepositions do not do well in the last column or row, as those positions will end a line. Of course, rules are meant to be broken, and in poetry, it is not uncommon for the end of one line to continue onto the next.

The poems in Figures 4 and 9 illustrate how an initial word placement can influence a poem. Both poems were inspired by a desire to reverse a message from one stanza to the next. To help achieve this, each poem has a negation in one of the central cells of the square, *not* in the poem from Figure 4 and *never* in the poem from Figure 9. This places each of these words in the centre of two lines. In the poem

from Figure 9, the placement of the word *never* generates the contrasting lines *End never swift* and *Time will never wait*.

Incremental improvements: Once a passable poem is constructed with some lines that work, try improving the poem one word at a time. For example, suppose while constructing the poem from Figure 10 we arrive at the following poem.

*Understand **hope** desire*

Beyond infinite need

Measures heart's love

Understand heart's need

Infinite desire measures

*Love beyond **hope***

We can consider replacing the word *hope* with *reason* keeping in mind that the word change must work in both positions in the stanza form. If this is an overall improvement, the swap is made and we arrive at a better poem. This process can then be repeated until the poet is satisfied. Don't be afraid to abandon the entire poem and start again if what you have feels truly unworkable. I found this stage of composition was greatly facilitated by software that automatically generates the stanza form using words typed into the square. I have written a Python program to do this for either a 3 by 3 or a 4 by 4 Graeco-Latin square poem. Readers are welcome to contact me to request a copy of the software.

Flexibility of the square: Once you have a satisfactory poem, you may be able to make further improvements by manipulating the Graeco-Latin square. New Graeco-Latin squares can be generated from an existing square by permuting its rows or columns. Such permutations have the potential to create nonsense out of a sensible poem, since they will scramble the word types. However, we can also permute the symbols (letters or numbers) in the square. The effect of this type of permutation is to permute entire lines of the stanza form, which is sometimes desirable. For example, we can easily reorder the first 2 lines of the second stanza of the poem.

Time will pass slowly,

Hours still wait,

End never swift

Time will never wait

Still slowly end

Swift hours pass

If we swap the A's and B's in the original Graeco-Latin square we will still have a Graeco-Latin square. If we also leave the word positions unchanged this reverses the first two lines of the second stanza, since the words paired with A generate the stanza's 1st line and the words paired with B generate the 2nd. The original square for this poem is shown in Figure 9. The new square obtained from swapping the A's and B's is shown below in Figure 11. The corresponding stanza form with the reordered lines is given to the left of Figure 11.

Time will pass slowly,
Hours still wait,
End never swift

Still slowly end
Time will never wait
Swift hours pass

1B Time will	2A Still	3C Swift
2C Hours	3B Never	1A Slowly
3A End	1C Pass	2B Wait

Figure 11: *Permuting symbols to reorder lines.*

The order of the stanzas can also be reversed by transposing the matrix and switching the letters and numbers within the square.

Sleight of hand: Finally, a successful Graeco-Latin square poem should leave the impression of a poetic sleight of hand. For the reader who understands how the poem is constructed, it might seem extraordinary that the second stanza comes together and is readable. The poet can take advantage of this. Suppose we are faced with a choice between two words, such as the word *hope* and the word *reason* in the poem from Figure 10. It might be argued that the word *hope* works better in the 1st stanza, when we compare the lines *Understand hope desire* versus *Understand reason desire*. However, when we compare the lines from the 2nd stanza we have *Love beyond hope* versus *Love beyond reason*. This time we can argue that the word *reason* is preferred. When faced with such a compromise, choose in favour of the second stanza, since that is where the reader expects the poem to fall apart. In fact, it is not a bad strategy to begin by writing a strong last line or two for the second stanza and then work towards improving the poem by applying the *incremental improvement* technique.

The above techniques have been presented in a logical order, giving a rough idea of the stage at which they are most useful in the process. However, while writing these poems I went back-and-forth between these techniques trying to find a poem that works. Despite the challenges, I encourage anyone who is interested in writing a Graeco-Latin square poem not to feel daunted by the task. Once the poet develops a feeling for how word swaps effect the poem, they might find that writing a Graeco-Latin square poem is not much more difficult than writing a poem with a strict rhyming scheme.

References

- [1] L. Lajeunesse. "Poetry puzzles." *Bridges Conference Proceedings*, Stockholm, Sweden, July 24 – 28, 2018, pp. 645–648.
<https://archive.bridgesmathart.org/2018/bridges2018-645.html>
- [2] B. Isett. Grid Poems. <https://gridpoems.com/>
- [3] S. Glaz. "Mathematical pattern poetry." *Bridges Conference Proceedings*, July 25 – 29, 2012, pp. 65–72.
<https://archive.bridgesmathart.org/2012/bridges2012-65.pdf>
- [4] L. Euler. "Recherches sur une Nouvelle Espece de Quarres Magiques." *Opera Omnia*, vol. 7, Ser 1, 1782, pp. 291–392. <http://eulerarchive.maa.org/index.html>
- [5] R.C. Bose, S.S. Shrikhande and E.T. Parker. "Further results on the construction of mutually orthogonal Latin squares and the falsity of Euler's conjecture." *Canadian Journal of Mathematics*, vol. 12, 1960, pp. 189–203.
- [6] H. Mathews and Alastair Brotchie (ed.). *Oulipo Compendium*. Atlas Press, 2011, pp. 174–177.
- [7] G. Perec. *Life: A User's Manual*. Translated by D. Bellos. David R. Godine, 2009.
- [8] G. Perec. *Cahier des charges de La Vie mode d-emploi*. CNRS, 1995.