

A MATHEMATICAL AND COMPUTER-BASED ALGORITHMIC METHOD FOR CONSTRUCTING ODD-ORDER EQUAL-SUM MATRIX NUMBER PUZZLES FROM ARITHMETIC SEQUENCES.

R.M.N.B. Rathnayake

Department of Information Technology, Faculty of Social Sciences and Languages,

Sabaragamuwa University of Sri Lanka, Sri Lanka

Abstract: It is important to say that number puzzles and equal sum matrices have been highly researched because of their strong connections to mathematics, pattern recognition, and problem solving by means of algorithms. However, a lot of methods of design of such matrices involve mathematical formulas and approaches that can be challenging for those who lack necessary mathematical knowledge. The goal of this research is to introduce the technique of design of Matrix Number Puzzles of odd orders based on arithmetic progressions where the sum of all rows, columns, and two diagonals must be the equal. Design Science Research Methodology (DSRM), an algorithm development, and mathematical validation methodology are selected for this research. Construction involves Straight-Line Left-Angle Procedure (SLLAP) and Straight-Line Right-Angle Procedure (SLRAP) processes in generating the embossed matrix pattern. The other values which are not part of the embossed matrix are then organized by using of the Interchanging Values and Placing Them in the Corresponding Locations (IVPTCL) process. The process is demonstrated by the use of odd-order square matrices of (3×3) , (5×5) , (7×7) , (9×9) , and (11×11) . In the finished matrices, it can be observed that the construction process being discussed is able to generate equal sums of the selected arithmetic sequences in all rows, columns, and main diagonals. Furthermore, the process performances as the foundation for implementing the process in a computer program in which the input parameters are the order of the matrix, first value, and common difference using the programming language of Python.

Keywords: Square matrix of odd order, arithmetic series, Matrix Number Puzzle, SLLAP, SLRAP, IVPTCL, equal-sum matrix, algorithm design, mathematics.

INTRODUCTION

Number puzzles have been considered significant in the field of recreational mathematics due to their combination of logic, patterns, numbers, thinking, and problem-solving in a structured mathematical environment. Number puzzles such as Sudoku and magic squares involve the discovery of the relationship between the numbers, which satisfy the given set of criteria. For instance, magic squares are known as puzzles involving the placement of numbers on a square grid such that the total of numbers along every row, every column, and along both diagonals are the equal (Block & Tavares, 2009; Sesiano, 2019). This has made equal sum numerical arrangements a subject of constant mathematical research and a basis of systematic puzzle formation.

The traditional approach for solving number puzzles is pattern identification, deduction, number computation, and the process of trial-and-error. While these methods are very useful when dealing with a simple numerical setup, it might get more complicated when dealing with a bigger structure of puzzle. Thus, computational methods are becoming increasingly popular in finding solutions for number puzzles. Various previous studies have addressed issues concerning the use of algorithmic, constraint, search and optimization methods in solving and generating numerical puzzles such as Sudoku and other number structures (Maji et al., 2013; Soto et al., 2013).

The evolution of information technologies has reinforced the bond between mathematics and computer-based problem solving even more. Mathematical relations, which require repetitive manual calculations in order to be formulated, can be expressed using algorithms and programmed on the computer. Modern scientific works prove that magic square generation is a computable process, and special algorithms for construction of matrices meeting certain numerical requirements can be used (Pamplona et al., 2026). In mathematics, there are construction techniques designed especially for construction of odd-order magic squares (Chan et al., 2014). However, some existing mathematical construction

techniques use formulas, notation and certain procedures that can be quite difficult to comprehend for people without profound mathematical knowledge.

In this regard, the current study specifically deals with the formulation of Matrix Number Puzzles of odd order through arithmetic sequences. An arithmetic sequence is studied based on its matrix order (n), first element (a), and common difference (d). The basic problem studied in this research is that how the elements of such an arithmetic sequence can be organized in an odd-order ($n \times n$) square matrix in such a manner that the sum of elements of each row, each column, and both diagonals should be equal to a common summing value. In this research, emphasis is specifically placed on finding this required summing value through diagonal summing formula and then organizing the arithmetic sequence elements to make the equal sum throughout the matrix.

In order to solve this problem, this study employs a visual and procedural construction technique aimed at lessening the reliance on complex mathematical equations in constructing the matrix arrangements. In this approach, the Straight-Line Left-Angle Procedure (SLLAP) and Straight-Line Right-Angle Procedure (SLRAP) are used. Both the procedures are used collectively as SL-LAP and RAP method to spot particular cells and construct an embossed matrix. In the construction process of the matrix, some numerals will stay in the embossed matrix, while others will be placed outside its boundaries. Those values outside the embossed matrix are then placed in the unoccupied positions in the matrix using the Interchanging Values and Placing Them in the Corresponding Positions (IVPTCP) technique.

The construction technique has been explained step-by-step in case of (3×3) , (5×5) , (7×7) , (9×9) and (11×11) odd-ordered square matrices. Various other arithmetic progressions can also be examined by changing the first term and common difference so as to test if the construction rules apply for such cases as well. The sum of all the rows, columns, and two main diagonals of the generated matrix has been compared with the common sum of the sequence. Mathematical proof has also been included in the research.

Besides, along with the manual and mathematics construction technique, there will be an attempt to convert the new technique into the algorithm to be programmed using the Python language. The algorithm will receive as input parameters the size of the matrix (n), where n must be an odd number, the first number (a), and the common difference between two consecutive terms in the series (d). Then, it will be possible to produce the arithmetic sequence that is needed for constructing the odd-order matrix and check the sums of rows, columns, and diagonals. The combination of mathematical calculations and computing can provide an easier way to create Matrix Number Puzzles for people without profound mathematical knowledge.

Consequently, the main contribution of this study can be seen as the creation of an organized method for creating odd order equal sum Matrix Number puzzles using arithmetic series. Through the use of diagonal summing, systematic methods for construction of lines, repositioning of values, mathematical analysis, and computational application, the study aims at establishing a link between recreational mathematics and information technology.

The research problem statement is as follows: What would be a way to arrange the terms of an arithmetic progression in an odd-order square matrix in such a way that the sum of all rows, columns, and both diagonals will be the equal and the procedure will be easy enough not to require the application of mathematical formulas from users?

LITERATURE REVIEW

Rathnayake (2026) developed and validated a matrix diagonal summation method for odd-order arithmetic square matrices. The study focuses on arranging the elements of odd-order square matrices formed from consecutive terms of an arithmetic sequence so that the sums of the rows, columns, and principal diagonals satisfy the required equal-sum property. The proposed method provides a systematic approach to constructing such matrices and demonstrates its applicability to odd-order arithmetic square matrices. The study contributes a mathematical technique that can be used for further investigations into structured matrix arrangements and rows, columns, and principal diagonals equal-sum properties.

Number puzzles fall under the category of recreational mathematics since they involve numerical relations, logical thinking, patterns, and structured problem solving. Various types of number puzzles come with different mathematical requirements when arranging the numbers. In particular, Sudoku and magic squares are common puzzles whose solution requires the fulfillment of certain structural and numerical conditions. The study of magic squares through history shows that there have been efforts to arrange numbers into structures where all sums are equal, for many years (Sesiano, 2019).

According to Block and Tavares (2009), magic squares and their properties have been broadly discussed. The authors describe a construction where the numerical items are arranged in a square such that the summation of the items is in agreement with the predetermined requirements. The main feature of a classical magic square is that the summation of

the items in every row, in every column and on both principal diagonals is the same. The property of the equal summation is directly related to the present study as the Matrix Number Puzzle will also be created based on the requirement of having the same summation for all rows, columns and principal diagonals.

While number puzzles have traditionally been approached using human reasoning, growing puzzle size and complexity may make it increasingly difficult to generate and solve them manually. This led to the investigation of computational methods that convert the process of solving puzzles into a well-defined algorithm. This becomes a very critical starting point for the current research endeavor to integrate mathematics and computerized procedure for odd order Matrix Number Puzzles.

The use of computational techniques on number games is proof of how mathematical rules can be expressed through algorithms. For instance, Maji et al. (2013) conducted research on an algorithm for generation of only necessary permutations in solving Sudoku games. This shows how the computational process can be simplified by coming up with relevant permutations instead of testing all combinations. This is just one of many examples of the importance of coming up with an algorithm based on the features of a math game.

Likewise, Soto et al. (2013) have suggested a method that was developed by the integration of the AC3 algorithm for processing constraints with the tabu search technique for resolving the Sudoku problem. This shows that structured constraint reasoning and computational search can be used together in the resolution of mathematical puzzles. It is important to note that although Sudoku is not at all similar to the Matrix Number Puzzle used in the present study, this research area proves that the number puzzles can be solved through computation.

The computational work done before is significant in this research since it shows the shift from puzzle-solving processes that were manually done to algorithmically repeatable processes. Nevertheless, the Sudoku-based approaches have been developed to find out the values that meet the constraints of the puzzle. In contrast, the present research deals with a completely different problem because it does not solve an incomplete Sudoku-like puzzle but builds a matrix of an odd order from a specified arithmetic progression so that each line, column, and main diagonal would give an equal sum. Hence, while the past computational puzzle-solving research offers a very valuable methodological base, the special matrix-building approach is necessary for this study.

Magic squares are very important for current research due to their property of equal sums. According to Block and Tavares (2009) and Sesiano (2019), magic squares have had a long history of being researched in mathematics, and many techniques have been found to position numerals in order to meet the condition of equal sums. The basic mathematical relationship between rows, columns, and main diagonals creates the foundation for further investigation.

The construction of odd-order magic squares using mathematics has been another area of interest for researchers. Chan et al. (2014) provided a mathematical construction of regular magic squares of odd orders, showing that odd-order matrices have structural characteristics that help construct them mathematically. This is especially pertinent to the current study as it has limitations regarding odd-order square matrices, including (3x3), (5x5), (7x7), (9x9), and (11x11). It thus proves the mathematical premise that odd-order equal-sum squares can be constructed mathematically rather than randomly.

In more recent times, however, Pamplona et al. (2026) looked at magic square construction as an integer constraint satisfaction problem and described an efficient way to construct such magic squares. This study shows that magic square construction is expressible algorithmically and computational algorithms can be devised to create matrices meeting the necessary equality conditions. This is relevant to the current study since both studies deal with matrix construction systematically and not just the solution of some pre-existing puzzle.

However, there is a difference between the current research and the existing magic squares as the current research specifically focuses on arithmetic sequences. Traditional magic square formulas often use consecutive numbers, but the current matrix number puzzle takes into consideration arithmetic sequences that have their first terms (a) and common differences (d). Therefore, some examples of such arithmetic sequences include (1,2,3,4,...), (2,4,6,8,...), and (3,7,11,15...), among others. Therefore, the objective of the current research is to examine how the principles of equal sum odd order matrices can be applied to a general arithmetic sequence.

A third theme noted in the literature is the increased adoption of algorithms for translating mathematical relations into computational operations. The works of Maji et al. (2013) and Soto et al. (2013) illustrate this concept using puzzles, whereas Pamplona et al. (2026) have employed computational logic to generate magic squares. Overall, these examples suggest that the algorithms can serve as a link between mathematical conditions and their implementation.

Nevertheless, the construction approaches in mathematics are still dependent on the user's ability to comprehend special notation, formulas, constraints, or even procedure. This research a significant point for users who are interested in number puzzles but have little mathematical knowledge. In this case, the use of a procedure which visually describes the construction process and shows the necessary movements in numbers through procedures becomes an alternative way of equal-sum matrix construction.

This research attempts to deal with this problem by using the Straight-Line Left-Angle Procedure (SLLAP) and the Straight-Line Right-Angle Procedure (SLRAP). These two procedures are used together as the SL-LAP and RAP approach in order to find out and create the embossed matrix pattern. The values outside the embossed matrix pattern will be put in the corresponding positions by using the Interchanging Values and Placing Them in the Corresponding Locations (IVPTCL). The uploaded research illustrates these procedures step by step from (3x3), (5x5), (7x7), (9x9), and (11x11) odd matrices.

This is significant because there is a way to transform the mathematics behind the solution into a visual construction process. The idea is not to have the user calculate things at each step, but to have them identify lines, positions, and rules for switching values that can then be turned into an algorithmic computer program. This is aligned with computational work more generally, as it shows how math puzzles can be transformed into algorithms.

RESEARCH GAP

The literature review indicates a significant body of knowledge on three interconnected areas: mathematics of magic squares and their history (Block & Tavares, 2009; Sesiano, 2019), mathematical approaches to the construction of odd order magic squares (Chan et al., 2014), and computer methods for solving or constructing numerical puzzles (Maji et al., 2013; Pamplona et al., 2026; Soto et al., 2013). This research demonstrates that line-method techniques and systems involving equal sums can be investigated using mathematical principles, while computer algorithms can assist in solving and constructing such puzzles.

Yet, among the existing literature surveyed in this research, there is little discussion of an easily understandable, linear method for organizing the numbers of an arithmetic progression in odd-order square matrix in such a way that each row, column, and the two principal diagonals have a similar sum. The current literature mainly focuses on the construction of magic squares, computationally oriented formulations with constraints, or solving other numeric puzzles. In addition, none of these approaches attempts to provide a method for construction that is easy to understand for those without mathematics background.

This research fills this gap by presenting a methodology that includes the use of the Diagonal Summation Method, SLLAP, SLRAP, the construction procedure of SL-LAP and RAP, as well as IVPTCL technique. All these methodologies are used for constructing and completing odd-order Matrix Number Puzzles in the with the arithmetic sequences. Finally, this mathematical procedure is coded in Python programming language in order to take n , a , and d as input data values and construct the necessary equal-sum matrix. Thus, this study expands the existing literature about magic square construction by concentrating on arithmetic sequence based odd-order matrices and the computational aspect of the process.

METHODOLOGY

The current research is based on the Design Science Research Methodology (DSRM), an algorithm development, and mathematical validation methodology. This is because the research revolves around designing, developing, implementing, and validating the computation and mathematics solution to constructing Matrix Number Puzzles of odd order from arithmetic progression. In this methodology, the research makes use of algorithm development and mathematical validation, with the implementation of a computerized program.

Methodological approach involves the analysis of mathematical properties of square matrices based on an arithmetic sequence. Analysis takes into account three key parameters: order of a matrix (n), first element (a), and common difference (d). These parameters are needed in order to establish the necessary numerical correlations which will be needed to find the common sum and arrange arithmetic sequence elements into odd-order square matrix.

In light of the found mathematical dependencies, Diagonal Summation Approach is developed as the main approach to construction of the mathematical model. The essence of the approach is in defining a set of rules for the distribution of the members of the arithmetic progression in the matrix, according to which the sums of the rows, columns, and two main diagonals are equal to the same number the diagonal sum. The correctness of the proposed approach is mathematically proved using the principle of mathematical induction.

After the mathematical derivation and validation, the suggested procedure is coded into a computer algorithm by employing Python programming language. The code is developed in such a way that the values of (n), (a), and (d) are taken from the user, the arithmetic sequence is generated based on the given parameters, an odd order matrix is developed, the suggested arrangement procedure is applied, and the sum of the elements is calculated.

In order to test the validity of the suggested procedure, the devised program is evaluated through various odd ordered square matrices, which include (3x3), (5x5), (7x7), (9x9), and (11x11) matrices, along with various arithmetic sequences. In each of these scenarios, the resulting row totals, column totals, and main diagonal totals are compared with the theoretical diagonal total summation value (Rathnayake, R. M. N. B. (2026)). Consistency in these scenarios is taken into account to check the validity of the suggested procedure.

Apart from the mathematical and functional validations, the behavior of the algorithm is studied computationally using various dimensions of matrices. This implies that the methodology employed includes mathematical analysis, mathematical modeling, algorithm design, mathematical proofs, software development, experimentation and computation. From this methodology, the study has been able to come up with both a theoretical framework as well as computational technique for construction and validation of odd-order Matrix Number Puzzles with arithmetic sequences.

RESULTS AND DISCUSSION

Number puzzles may be researched using systematic and proper mathematical approaches. In this research, the method used is tested on odd-order square matrices of the dimensions (3x3), (5x5), (7x7), (9x9), (11x11)...(nxn) and other higher odd-order matrices. The aim of the test is to solve the research gap found. The method is examined using several arithmetic sequences, including 1, 2, 3, 4, 5, 6, ...; or 2, 4, 6, 8, ...; or 3, 7, 11, 15, ...; as well as other arithmetic sequences defined by different first terms and common differences. The validity of the proposed approach is further established through mathematical proof based on the principle of mathematical induction.

Various research studies have been conducted to identify techniques for designing square matrices of odd-order in which the sum of all rows, columns, and both diagonals is identical. Nevertheless, a large number of these techniques make use of complicated mathematical equations and formulas or complicated design strategies that might be difficult to understand by people who are not well versed with mathematics. The present research work aims at presenting a simple line-based strategy for designing these matrices.

Terms and Notation in a Matrix:

Rows and Columns: The horizontal and vertical arrangements of elements in a matrix are referred to as **rows** and **columns**, respectively [Figure 1].

Order/size of a Matrix: The order/size of a matrix is written as (**m x n**), read as “(m) by (n),” where (**m**) represents the number of rows and (**n**) represents the number of columns [Figure 1].

Elements of a Matrix: The individual values or entries contained within a matrix are called **elements**. Each element is generally identified according to its row and column positions. For example, an element may be denoted by (**a_{ij}**), where (**i**) represents the row number and (**j**) represents the column number [Figure 1].

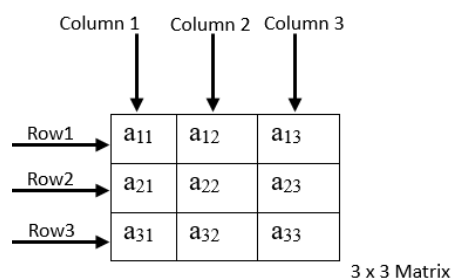


Figure 1: Core Terms and Notation in a Matrix

Straight-Line Left-Angle Procedure (SLLAP) for an Odd-Order Square Matrix:

In this study, a new method, referred to as the **SLLAP**, is introduced for constructing odd-order square matrices. The method is demonstrated using matrices of orders (3x3), and (5x5) to illustrate its application and effectiveness [Figure 2][Figure 3].

1	2	3
4	5	6
7	8	9

(3×3) Matrix

with SLLAP

Figure 2: Straight-Line Left-Angle Procedure (SLLAP) for (3×3) Matrix

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

(5×5) Matrix with SLLAP

Figure 3: Straight-Line Left-Angle Procedure (SLLAP) for (5×5) Matrix

The **SLLAP** is used to separate and systematically arrange the numbers between two straight lines within each matrix. In this study, the applicability of the SLLAP is extended to larger odd-order square matrices, including (7×7) , (9×9) , (11×11) , ..., $(2k + 5) \times (2k + 5)$ matrices, in order to demonstrate the general applicability of the proposed method, where (k) is a positive integer ($(k = 1, 2, 3, \dots)$).

Straight-Line Right-Angle procedure (SLRAP) for an Odd-Order Square Matrix:

In this study, a new method, referred to as the **SLRAP**, is introduced for constructing odd-order square matrices. The method is demonstrated using matrices of orders (3×3) , (5×5) to illustrate its application and effectiveness [Figure 4, Figure 5].

1	2	3
4	5	6
7	8	9

(3×3) Matrix with

SLRAP

Figure 4: Straight-Line Right-Angle procedure (SLRAP) for (3×3) Matrix

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

(5×5) Matrix with SLRAP

Figure 5: Straight-Line Right-Angle procedure (SLRAP) for (5×5) Matrix

The **SLRAP** is used to separate and arrange the numbers between two straight lines within each matrix. In this study, the applicability of the SLLAP is extended to larger odd-order square matrices, including (7×7) , (9×9) , (11×11) , ..., $(2k + 5) \times (2k + 5)$ matrices, in order to demonstrate the general applicability of the proposed method, where (k) is a positive integer ($(k = 1, 2, 3, \dots)$).

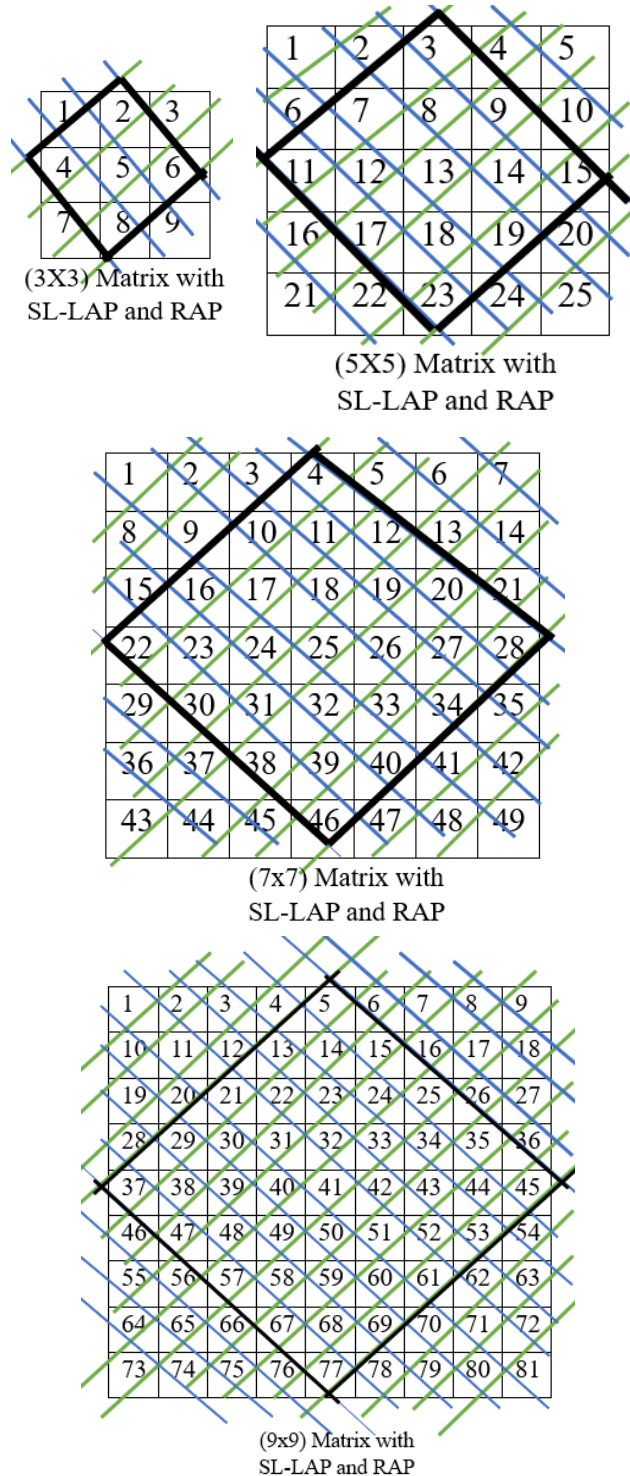
Straight-Line Left-Angle Procedure and Right-Angle procedure (SL-LAP and RAP) for an Odd-Order Square Matrix:

In this study, a new construction method that integrates the **SLLAP** and the **SLRAP** is introduced for constructing

embossed odd-order square matrices. These procedures, referred to collectively as the **SL-LAP and RAP** method, are applied together to achieve the required matrix arrangement. The proposed method is demonstrated using (3×3) , (5×5) , (7×7) , (9×9) , and (11×11) matrices to illustrate its application, effectiveness, and consistency.

Next, the **SL-RAP method** is represented by green lines, while the **SL-LAP method** is represented by blue lines within the odd-order square matrix. Then newly **embossed matrices** of the same order are displayed beneath the black line for further analysis.

The newly embossed (3×3) , (5×5) , (7×7) , (9×9) , and (11×11) matrices are presented below to facilitate the understanding and identification of embossed matrices of the same order [Figure 6].



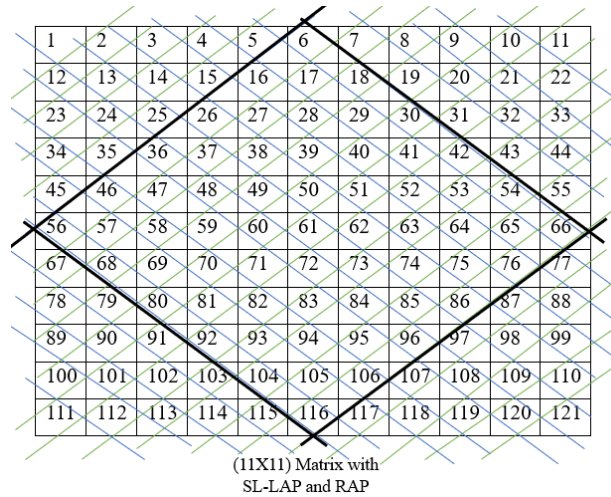


Figure 6: Newly embossed (3x3), (5x5), (7x7), (9x9), (11x11) Matrices

• 3x3 Newly Embossed Matrix:

The problem considers a (3x3) matrix containing the arithmetic sequence 1,2,3,4,5,6,7,8,9. These values must be arranged so that the sums of all rows, all columns, and both principal diagonals are equal to 15. This common summation value is determined from the equality of the two principal diagonal sums, where $TLBR = TRBL = 15$. (Rathnayake, R. M. N. B. (2026)).

When the newly embossed (3x3) matrix is considered [Figure 6], several positions within the matrix remain unassigned. Specifically, no values are initially placed at $L[1,2]$, $L[2,1]$, $L[2,3]$, and $L[3,2]$. In contrast, values are assigned to the remaining positions as follows: $L[1,1]=4$, $L[1,3]=2$, $L[2,2]=5$, $L[3,1]=8$, and $L[3,3]=6$. In addition, total four circled values 1,3,7, and 9 are positioned outside the boundaries of the newly constructed embossed matrix [Figure 7].

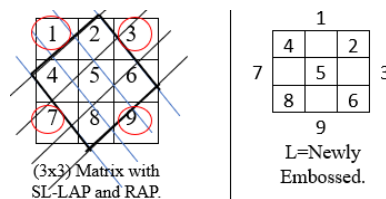


Figure 7: Newly embossed and (3x3) Matrix with side values

To obtain the final arrangement of the (3x3) embossed matrix, the outside values 1,3,7, and 9 are systematically repositioned into the unassigned cells of the matrix. The objective is to complete the matrix so that the sums of all **rows**, **columns**, and **both principal diagonals** are equal to the same common summation value. (Rathnayake, R. M. N. B. (2026)) [Figure 8].

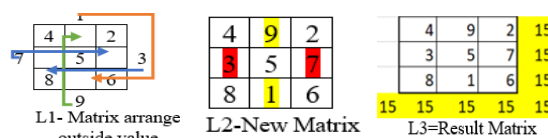


Figure 8: Matrix with outside value, new arranged and Resulted matrix

- In the **second column**, the outside available values are 1 and 9. These values are inserted into the embossed matrix by using **IVPTCL** technique. Accordingly, the values are assigned as $L[1,2]=9$ and $L[3,2]=1$. (see L2 in Figure 8).
- In the **second row**, the outside available values are 3 and 7. These values are inserted into the embossed matrix by using **IVPTCL** technique. Accordingly, the values are assigned as $L[2,1]=3$ and $L[2,3]=7$. (see L2 in Figure 8).

- Now, consider the resulting matrix and calculate the sums of all **rows, columns, and principal diagonals** to verify that they are equal to the same common summation value. (see L3 in Figure 8).

• (5x5) Newly Embossed Matrices:

The problem considers a (5x5) matrix containing the arithmetic sequence 1,2,3,...25. These values must be arranged so that the sums of **all rows, all columns, and both principal diagonals** are equal to 65. This common summation value (65) is determined from the equality of the two principal diagonal sums, where $TLBR = TRBL = 65$ (Rathnayake, R. M. N. B. (2026))

Similarly, the **IVPTCL technique** is applied to a (5×5) odd-order square matrix to arrange the elements according to the selected arithmetic sequence, as demonstrated below.

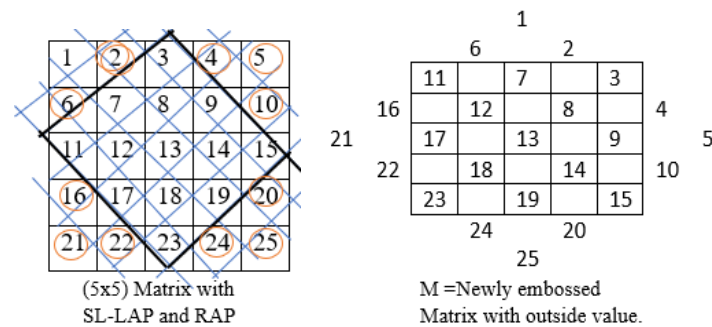


Figure 9: Newly embossed Matrix with SL-LAP and RAP and with the outside value

The newly embossed (5x5) matrix is constructed using the **SL-LAP and RAP technique**. The assigned values are $M[1,1]=11$, $M[1,3]=7$, $M[1,5]=3$, $M[2,2]=12$, $M[2,4]=8$, $M[3,1]=17$, $M[3,3]=13$, $M[3,5]=9$, $M[4,2]=18$, $M[4,4]=14$, $M[5,1]=23$, $M[5,3]=19$, and $M[5,5]=15$ [Figure 9].

However, several positions in the newly embossed matrix remain unassigned, including $M[1,2]$, $M[1,4]$, $M[2,1]$, $M[2,3]$, $M[2,5]$, $M[3,2]$, $M[3,4]$, $M[4,1]$, $M[4,3]$, $M[4,5]$, $M[5,2]$, and $M[5,4]$ [Figure 9].

In addition, total twelve (12) values **6, 1, 2, 4, 5, 10, 16, 21, 22, 24, 25, and 20** are positioned outside the boundaries of the newly embossed matrix and are represented using circles [Figure 9]. These outside values are subsequently used to complete the unassigned positions of the matrix by applying the technique of **IVPTCL**.

Considering the newly embossed (5x5) matrix, the values positioned outside the matrix can be grouped according to their corresponding columns and rows. The column-related value sets are **1, 2, 6** and **20, 24, 25**, while the row-related value sets are **16, 21, 22** and **4, 5, 10**. These values are then placed into the unassigned matrix positions by following the proposed procedure, as described in the following steps:

- In the **second column**, the outside values are 6 and 24. These values are inserted into the newly embossed matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $M[1,2]=24$ and $M[5,2]=6$, as illustrated in M1 [Figure 10].
- In the **third column**, the outside values are 1 and 25. These values are inserted into the newly embossed matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $M[2,3]=25$ and $M[4,3]=1$, as illustrated in M1 [Figure 10].
- In the **fourth column**, the outside values are 2 and 20. These values are inserted into the newly embossed matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $M[5,4]=2$ and $M[1,4]=20$, as illustrated in M1 [Figure 10].

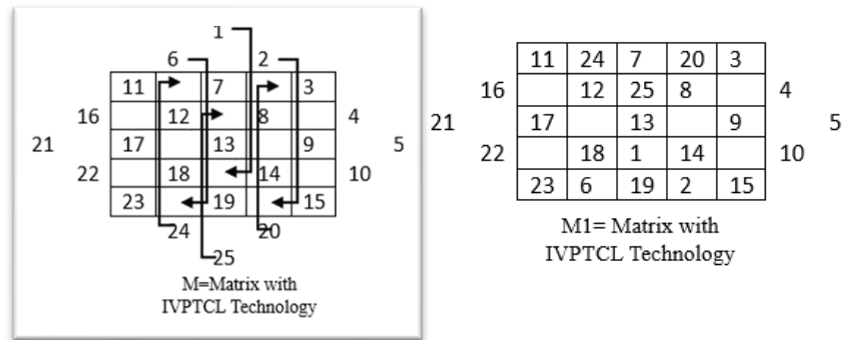


Figure 10:Matrix with IVPTCL Technology with outside values

- In the **second row**, the outside values are 16 and 4. These values are inserted into the matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $M[2,1]=4$ and $M[2,5]=16$, as illustrated in M1 [Figure 11].
- In the **third row**, the outside values are 21 and 5. These values are inserted into the matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $M[3,2]=5$ and $M[3,4]=21$, as illustrated in M1 [Figure 11].
- In the **fourth row**, the outside values are 22 and 10. These values are inserted into the matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $M[4,1]=10$ and $M[4,5]=22$, as illustrated in M2 [Figure 11].

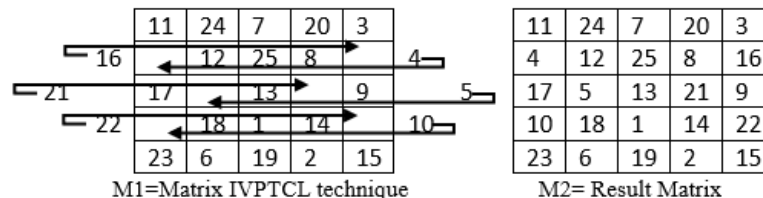


Figure 11:Matrix with IVPTCL Technology with outside values and Result Matrix

After filling the unassigned positions using the **IVPTCL technique**, the required solution to the previously stated problem is obtained. The completed matrix is presented below [Figure 12].

	11	24	7	20	3	65
	4	12	25	8	16	65
	17	5	13	21	9	65
	10	18	1	14	22	65
	23	6	19	2	15	65
65	65	65	65	65	65	65

Figure 12:Results Matrix After Applying the IVPTCL Technique

• (7x7) Newly Embossed Matrices:

The problem considers a (7x7) matrix containing the arithmetic sequence 1,2,3,...49. These values must be arranged so that the sums of **all rows**, **all columns**, and **both principal diagonals** are equal to 175. This common summation value (175) is determined from the equality of the two principal diagonal sums, where $TLBR = TRBL = 175$ (Rathnayake, R. M. N. B. (2026)).

Similarly, the newly embossed matrices for (7x7) and so on can be created using the SL-LAP and RAP technique. The (7x7) newly embossed matrices are as follows [Figure 13]:

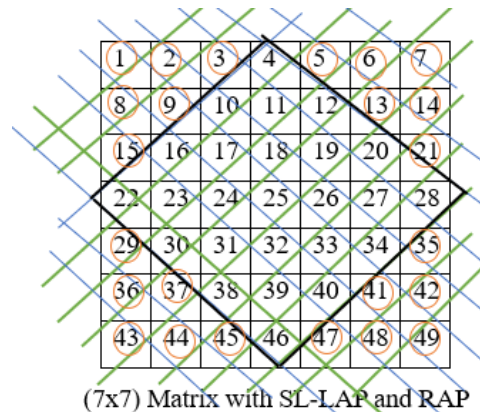


Figure 13: Newly embossed Matrix with SL-LAP and RAP and with the outside value

The newly embossed (7 $\times$ 7) matrix, denoted by N, is constructed using the **SL-LAB and RAP technique**. The assigned values are $N[1,1]=22$, $N[1,3]=16$, $N[1,5]=10$, $N[1,7]=4$, $N[2,2]=23$, $N[2,4]=17$, $N[2,6]=11$, $N[3,1]=30$, $N[3,3]=24$, $N[3,5]=18$, $N[3,7]=12$, $N[4,2]=31$, $N[4,4]=25$, $N[4,6]=19$, $N[5,1]=38$, $N[5,3]=32$, $N[5,5]=26$, $N[5,7]=20$, $N[6,2]=39$, $N[6,4]=33$, $N[6,6]=27$, $N[7,1]=46$, $N[7,3]=40$, $N[7,5]=34$, and $N[7,7]=28$ [Figure 14].

However, several positions in the newly embossed matrix remain unassigned. These positions are $N[1,2]$, $N[1,4]$, $N[1,6]$, $N[2,1]$, $N[2,3]$, $N[2,5]$, $N[2,7]$, $N[3,2]$, $N[3,4]$, $N[3,6]$, $N[4,1]$, $N[4,3]$, $N[4,5]$, $N[4,7]$, $N[5,2]$, $N[5,4]$, $N[5,6]$, $N[6,1]$, $N[6,3]$, $N[6,5]$, $N[6,7]$, $N[7,2]$, $N[7,4]$, and $N[7,6]$ [Figure 14].

In addition, total 24 values **1, 2, 3, 8, 9, 15, 5, 6, 7, 13, 14, 21, 35, 41, 42, 47, 48, 49, 29, 36, 37, 43, 44, and 45** are positioned outside the boundaries of the newly embossed matrix and are represented using circles. These outside values are subsequently used to fill the unassigned positions of the matrix according to the proposed arrangement procedure.

Now, consider the (7 $\times$ 7) matrix and group the outside values according to their corresponding columns and rows. The column-related value sets are **1, 2, 3, 8, 9, 15** and **35, 41, 42, 47, 48, 49**, while the row-related value sets are **29, 36, 37, 43, 44, 45** and **5, 6, 7, 13, 14, 21**. These values are inserted into the newly embossed matrix by applying the **IVPTCL technique to the unsigned cells**. [Figure 15].

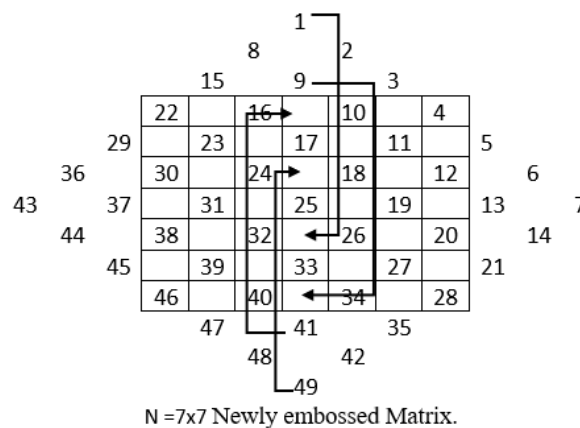


Figure 14: Matrix with IVPTCL Technology with outside values

- In the **fourth column**, the outside values sets are (1 and 9) and (41 and 49). These values are inserted into the newly embossed matrix by applying the **IVPTCL technique**. Accordingly, the values are assigned as $N[7,4]=9$ and $N[5,4]=1$, $N[1,4]=41$ and $N[3,4]=49$. as illustrated in N1 [Figure 15Error! Reference source not found.].

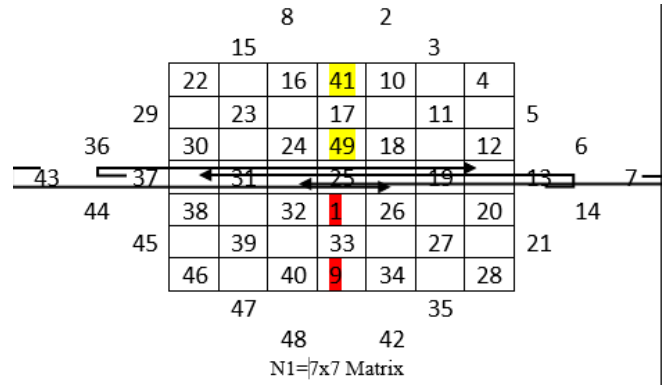


Figure 15: Matrix with IVPTCL Technology with outside values

- In the **fourth row**, the outside values sets are **(13 and 7) and (37 and 43)**. These values are inserted into the newly embossed matrix by applying the **IVPTCL** technique. Accordingly, the values are assigned as $N1[4,1]=13$ and $N[4,3]=7$, $N[4,7]=37$ and $N[4,5]=43$. as illustrated in N1 [Figure 15].

Based on the **IVPTCL technique**, all the outer value are systematically positioned in the rows and columns of the (7×7) matrix. The completed matrix is presented below [Figure 16]:

22	47	16	41	10	35	4
5	23	48	17	42	11	29
30	6	24	49	18	36	12
13	31	7	25	43	19	37
38	14	32	1	26	44	20
21	39	8	33	2	27	45
46	15	40	9	34	3	28

Figure 16: IVPTCL technique completed for (7×7) matrix

After filling the unassigned positions using the **IVPTCL technique**, the required solution to the previously stated problem is obtained, as presented below. [Figure 17]:

22	47	16	41	10	35	4	175
5	23	48	17	42	11	29	175
30	6	24	49	18	36	12	175
13	31	7	25	43	19	37	175
38	14	32	1	26	44	20	175
21	39	8	33	2	27	45	175
46	15	40	9	34	3	28	175
175	175	175	175	175	175	175	175

Figure 17: Results Matrix After Applying the IVPTCL Technique

- (9x9) Newly Embossed Matrices:**

The problem considers a (9×9) matrix containing the arithmetic sequence $1, 2, 3, \dots, 81$. These values must be arranged so that the sums of **all rows**, **all columns**, and **both principal diagonals** are equal to 369. This common summation value (369) is determined from the equality of the two principal diagonal sums, where $TLBR = TRBL = 369$ (Rathnayake, R. M. N. B. (2026)).

The newly embossed (9×9) square matrix, is constructed using the **SL-LAP and RAP techniques**. The assigned and unassigned values are indicated within the embossed matrix, while the remaining values are positioned outside the boundary of the newly embossed (9×9) square matrix, as illustrated below.

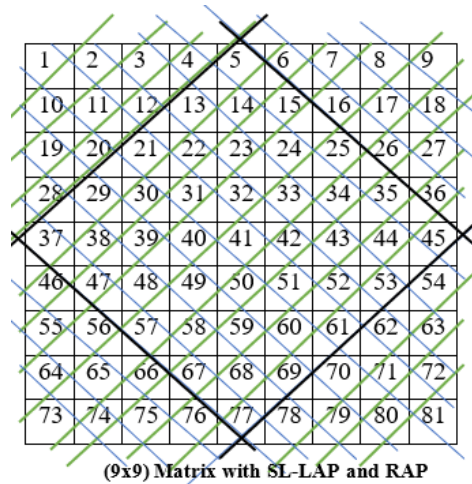


Figure 18: Newly embossed Matrix with SL-LAP and RAP and with the outside value

In addition, total 40 values are positioned outside the boundaries of the newly embossed matrix. These outside values are grouped according to their corresponding rows and columns [Figure 19]

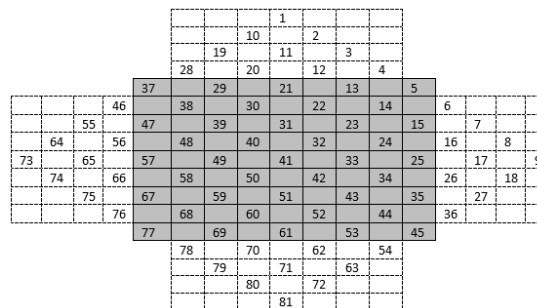


Figure 19: 9x9 Matrix with IVPTCL Technology with outside values

The twenty column-associated values are partitioned into two sets. The **first set** comprises 1, 2, 3, 4, 10, 11, 12, 19, 20, and 28, whereas the **second set** comprises 78, 79, 80, 81, 70, 71, 72, 62, 63, and 54. Subsequently, using the IVPTCL technique, the outer values are systematically positioned in the newly embossed matrix, following the same procedure used for the previous 5x5 or 7x7 order matrix. The resulting matrix is presented below. [Figure 20].

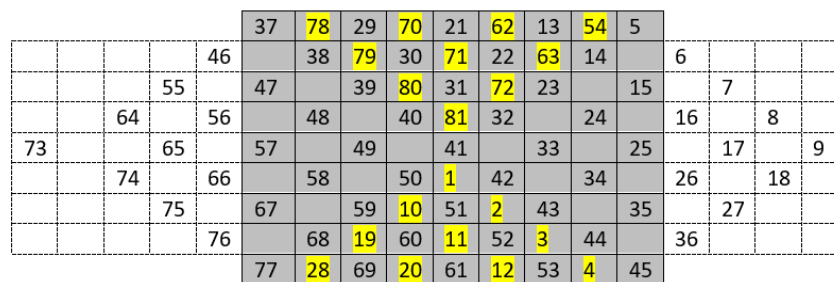


Figure 20: Filled the column sets of the Newly embossed 9x9 square matrix

Similarly, the twenty values associated with the rows are divided into two sets. The **first set** consists of 73, 74, 75, 76, 64, 65, 66, 55, 56, and 46, while the **second set** consists of 6, 7, 8, 9, 16, 17, 18, 26, 27, and 36. using the IVPTCL technique, the outer values are systematically positioned in the newly embossed matrix, following the same procedure used for the previous 5x5 or 7x7 order matrix. The resulting matrix is presented below [Figure 21].

			1			
		10		2		
	19		11		3	
28		20		12		4
37		29		21		5
6	38		30		22	14
47	7	39		31		23
16	48	8	40		32	64
57	17	49	9	41	73	33
26	58	18	50		42	74
67	27	59		51		43
36	68		60		52	
77		69		61		53
	78		70		62	
		79		71		63
			80		72	
				81		

Figure 21:Filled the rows sets of the Newly embossed 9x9 squire matrix

Based on the **IVPTCL technique**, all the outer value are systematically positioned in the rows and columns of the (9×9) matrix. The completed matrix is presented below [

Figure 22].

37	78	29	70	21	62	13	54	5
6	38	79	30	71	22	63	14	46
47	7	39	80	31	72	23	55	15
16	48	8	40	81	32	64	24	56
57	17	49	9	41	73	33	65	25
26	58	18	50	11	42	74	34	66
67	27	59	10	51	2	43	75	35
36	68	19	60	1	52	3	44	76
77	28	69	20	61	12	53	4	45

Figure 22:after filled rows and column using IVPTCL technique

After filling the unassigned positions using the **IVPTCL technique**, the required solution to the previously stated problem is obtained, as presented below [Figure 23].

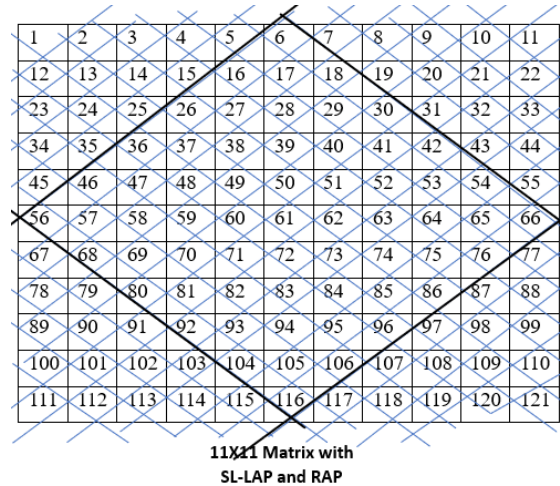
37	78	29	70	21	62	13	54	5	369
6	38	79	30	71	22	63	14	46	369
47	7	39	80	31	72	23	55	15	369
16	48	8	40	81	32	64	24	56	369
57	17	49	9	41	73	33	65	25	369
26	58	18	50	11	42	74	34	66	379
67	27	59	10	51	2	43	75	35	369
36	68	19	60	1	52	3	44	76	359
77	28	69	20	61	12	53	4	45	369
369	369	369	369	369	369	369	369	369	369

Figure 23:: Results Matrix After Applying the IVPTCL Technique

• 11x11 Newly Embossed Matrices:

The problem considers a (11×11) matrix containing the arithmetic sequence 1,2, 3,...121. These values must be arranged so that the sums of **all rows, all columns, and both principal diagonals** are equal to **671**. This common summation value (671) is determined from the equality of the two principal diagonal sums, where **TLBR = TRBL = 671** (Rathnayake, R. M. N. B. (2026))

After applying the **SL-LAP and RAP techniques**, a newly embossed (11×11) odd-order square matrix can be constructed. In this matrix, several values are positioned within the embossed matrix, while some values remain outside its boundaries. These outside values are subsequently used to complete the unassigned positions of the matrix according to the proposed arrangement procedure [Figure 24].



11x11 Matrix with
SL-LAP and RAP

Figure 24: Newly embossed Matrix with SL-LAP and RAP and with the outside value

In addition, a total of **60 value** is located outside the boundaries of the newly embossed (11×11) matrix. These outside values are subsequently grouped according to their corresponding rows and columns for further application of the proposed procedure [Figure 25].

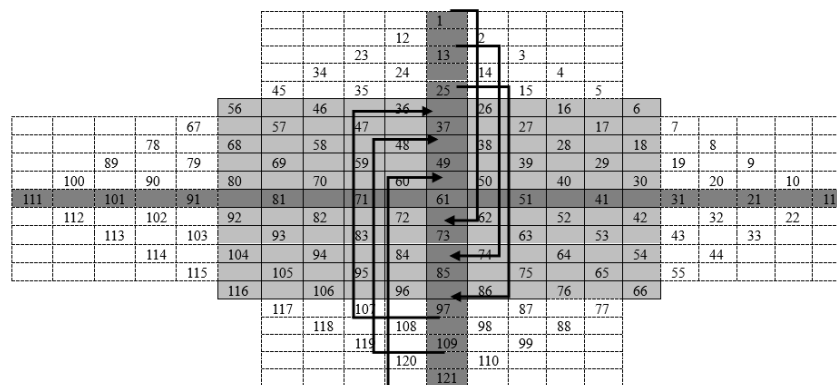


Figure 25: Matrix with IVPTCL Technology with outside values with change column

The thirty column-associated values are partitioned into two sets. The **first set** comprises **1, 2, 3, 4, 5, 12, 13, 14, 15, 23, 24, 25, 34, 35, and 45**, whereas the **second set** comprises **77, 87, 88, 97, 98, 99, 107, 108, 109, 110, 117, 118, 119, 120, and 121**. using the **IVPTCL technique**, the outer values are systematically positioned in the newly embossed matrix, following the same procedure used for the previous 7x7 or 9x9 order matrix. The resulting matrix is presented below [Figure 26].

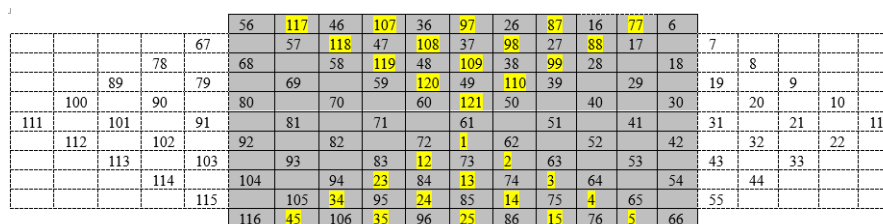


Figure 26: Matrix with IVPTCL Technology with outside column values

Similarly, the thirty values associated with the rows are divided into two sets. The **first set** consists of **67, 78, 79, 89, 90, 91, 100, 101, 102, 103, 111, 112, 113, 114, and 115**, while the **second set** consists of **7, 8, 9, 10, 11, 19, 20, 21, 22, 31, 32, 33, 43, 44, and 55**. using the **IVPTCL technique**, the outer values are systematically positioned in the newly

embossed matrix, following the same procedure used for the previous 7x7 or 9x9 order matrix. The resulting matrix is presented below [Figure 27].

				12	1	2			
		23			13		3		
	34		24		14		4		
	45		35		25		15		5
56		46		36		26		16	
7	57		47		37		27		17
68	8	58		48		38		28	18
19	69	9	59		49		39	89	29
80	20	70	10	60		50	100	40	90
31	81	21	71	11	61	111	51	101	41
92	32	82	22	72		62	112	52	102
43	93	33	83		73		63	113	53
104	44	94		84		74		64	114
55	105		95		85		75		65
116		106		96		86		76	
	117		107		97		87		77
		118		108		98		88	
			119		109		99		
				120		110			
					121				

Figure 27: Matrix with IVPTCL Technology with outside column values

Based on the **IVPTCL technique**, all the outer value are systematically positioned in the rows and columns of the (11×11) matrix. The completed matrix is presented below [Figure 28].

56	117	46	107	36	97	26	87	16	77	6
7	57	118	47	108	37	98	27	88	17	67
68	8	58	119	48	109	38	99	28	78	18
19	69	9	59	120	49	110	39	89	29	79
80	20	70	10	60	121	50	100	40	90	30
31	81	21	71	11	61	111	51	101	41	91
92	32	82	22	72	1	62	112	52	102	42
43	93	33	83	12	73	2	63	113	53	103
104	44	94	23	84	13	74	3	64	114	54
55	105	34	95	24	85	14	75	4	65	115
116	45	106	35	96	25	86	15	76	5	66

Figure 28: Results Matrix After Applying the IVPTCL Technique

After filling the unassigned positions using the **IVPTCL technique**, the required solution to the previously stated problem is obtained, as presented below. [Figure 29]

56	117	46	107	36	97	26	87	16	77	6	671
7	57	118	47	108	37	98	27	88	17	67	671
68	8	58	119	48	109	38	99	28	78	18	671
19	69	9	59	120	49	110	39	89	29	79	671
80	20	70	10	60	121	50	100	40	90	30	671
31	81	21	71	11	61	111	51	101	41	91	671
92	32	82	22	72	1	62	112	52	102	42	671
43	93	33	83	12	73	2	63	113	53	103	671
104	44	94	23	84	13	74	3	64	114	54	671
55	105	34	95	24	85	14	75	4	65	115	671
116	45	106	35	96	25	86	15	76	5	66	671
671	671	671	671	671	671	671	671	671	671	671	671

Figure 29: Results Matrix After Applying the IVPTCL Technique

Prove of IVPTCL, SL-LAP, and RAP Techniques:

Consider the arithmetic sequence 1,2, 3, 4,...etc. and construct an $(n \times n)$ square matrix using (n^2) consecutive terms of this sequence, where (n) is an odd positive integer. Using the **IVPTCL, SL-LAP, and RAP techniques**, arrange the elements of the matrix such that the sums of all rows, all columns, and both principal diagonals are equal.

First, demonstrated the validity of the **IVPTCL, SL-LAP, and RAP** techniques for the (3×3) square matrix using the first nine consecutive terms, 1,2, 3,...,9. Hence, those techniques are true for $(n=3)$.

Subsequently, apply the **IVPTCL, SL-LAP, and RAP** techniques to construct and verified for (5×5) , (7×7) , (9×9) , and (11×11) square matrices using (25), (49), (81), and (121) consecutive terms, respectively, from the same arithmetic sequence 1,2,3,4, ...etc. For each matrix order, it is demonstrated that the arrangement produced using the **IVPTCL, SL-LAP, and RAP** techniques yield equal sums for every row, every column, and both principal diagonals.

Finally, a general theorem is formulated for an arbitrary odd-order square matrix and proved using the **Principle of Mathematical Induction**, thereby establishing that the **IVPTCL, SL-LAP, and RAP techniques** are valid for square matrices of every odd order.

$n=3,5,7,9,11,\dots$ etc.

In particular, it is established that, for every positive integer (k) , the **IVPTCL, SL-LAP, and RAP** techniques can be applied to a $((2k+1) \times (2k+1))$ square matrix constructed from consecutive terms of the arithmetic sequence (1,2,3, ...etc.), such that all row sums, column sums, and the two principal diagonal sums are equal. Therefore, based on the principle of Mathematical induction, it is proved that the **IVPTCL, SL-LAP, and RAP** techniques are valid for square matrices of **all odd orders**.

more generally, that the construction remains valid for **any arithmetic sequence**, regardless of its first term (a) or common difference (d) , provided that the (n^2) elements are consecutive terms of that arithmetic sequence.

*As illustrative examples, the validity of the proposed technique is examined for four (04) odd-order square matrices. First, for $(n = 13)$, using an arithmetic sequence with first term $(a = 3)$ and common difference $(d = 3)$, the resulting common sum is **3,315**. Similarly, for $(n = 7)$, using an arithmetic sequence with first term $(a = 3)$ and common difference $(d = 5)$, the resulting common sum is **861**. Third, for $(n = 15)$, using an arithmetic sequence with first term $(a = 1)$ and common difference $(d = 3)$, the resulting common sum is **5,055**. Finally, for $(n = 3)$, using an arithmetic sequence with first term $(a = 5)$ and common difference $(d = 4)$, the resulting common sum is **63**.*

The Python program developed to implement the proposed matrix construction and validation of **IVPTCL, SL-LAP, and RAP** Techniques is presented below.

```
def generate_matrix(n, a, d):
    # n = order of the matrix (must be odd)
    # a = first term of the arithmetic sequence
    # d = common difference
    # Check whether n is a valid odd-order matrix
    if n < 3 or n % 2 == 0:
        raise ValueError("Matrix order must be an odd number: 3, 5, 7, 9, 11, ...")

    # Create an empty n x n matrix
    matrix = [[0 for _ in range(n)] for _ in range(n)]
    # There are n^2 values in the arithmetic sequence
    for k in range(n * n):
        # q identifies the group of n values
        q = k // n
        # t identifies the position inside the group
        t = k % n
        # Determine row and column positions
        # According to the arrangement pattern
        row = (q + (n + 1) // 2 + t) % n
        column = (n // 2 - q + t) % n
        # Generate the corresponding arithmetic-sequence value
        value = a + (k * d)
```

```
# Insert the value into the matrix
    matrix[row][column] = value
    return matrix
def calculate_common_sum(n, a, d):

#Calculate the required common/diagonal summation value.
    common_sum = n * (a + ((n * n - 1) // 2) * d)
    return common_sum
def verify_matrix(matrix):

#Calculate all row sums, column sums,
#TLBR diagonal sum, and TRBL diagonal sum.
    n = len(matrix)
    # Row sums
    row_sums = []
    for row in matrix:
        row_sums.append(sum(row))

# Column sums
    column_sums = []
    for column in range(n):
        total = 0
        for row in range(n):
            total += matrix[row][column]
        column_sums.append(total)

# Top-Left to Bottom-Right diagonal
    TLBR = 0
    for i in range(n):
        TLBR += matrix[i][i]
# Top-Right to Bottom-Left diagonal
    TRBL = 0
    for i in range(n):
        TRBL += matrix[i][n - 1 - i]
    return row_sums, column_sums, TLBR, TRBL
def display_matrix(matrix):

#Display the completed matrix clearly.
    print("\nCompleted Matrix:\n")
    for row in matrix:
        for value in row:
            print(f'{value:6}', end=" ")
        print()

# =====
# MAIN PROGRAM
# =====
print("ODD-ORDER MATRIX NUMBER PUZZLE")
print("-----")
# Input by the User
n = int(input("Enter odd matrix order n (3, 5, 7, 9, 11, ...): "))
a = int(input("Enter the first term a: "))
d = int(input("Enter the common difference d: "))
try:
    # Generate the matrix
    matrix = generate_matrix(n, a, d)
    # Calculate expected common summation
    common_sum = calculate_common_sum(n, a, d)
    # Display the matrix
```

```
display_matrix(matrix)
# Verify the matrix
row_sums, column_sums, TLBR, TRBL = verify_matrix(matrix)
print("\nArithmetic Sequence:")
sequence = [a + i * d for i in range(n * n)]
print(sequence)
print("\nExpected Common Summation Value =", common_sum)
print("\nRow Sums:")
for i, total in enumerate(row_sums, start=1):
    print(f"Row {i} = {total}")
print("\nColumn Sums:")
for i, total in enumerate(column_sums, start=1):
    print(f"Column {i} = {total}")
print("\nPrincipal Diagonal Sums:")
print("TLBR =", TLBR)
print("TRBL =", TRBL)
# Final validation
all_sums = row_sums + column_sums + [TLBR, TRBL]
if all(total == common_sum for total in all_sums):
    print("\nVALIDATION RESULT:")
    print("SUCCESS")
    print("All rows, columns, TLBR, and TRBL")
    print("have the same summation value =", common_sum)
else:
    print("\nVALIDATION RESULT:")
    print("The matrix does not satisfy the required condition.")
except ValueError as error:
    print("\nError:", error)
```

RESEARCH LIMITATION

While this suggested methodology offers an organized framework for generating odd order matrix number puzzles, there are certain constraints that should be noted within the current study.

To start with, the study is limited to odd order square matrices only. This methodology has been illustrated using odd ordered matrices like (3×3), (5×5), (7×7), (9×9), and (11×11). Hence, the current study does not prove the applicability of this approach to even order square matrices.

Secondly, the research is specifically concerned with arithmetic sequences. While different arithmetic sequences can be obtained by manipulating the initial term and the common difference, other types of numerical sequences like geometric and non-linear sequences have been excluded from this study.

Thirdly, while the proposed method is to be implemented using the Python programming language and while the computational performance has been cited as one of the evaluation factors, the current paper does not give details of any quantitative performance measure like execution time or memory usage for different orders of matrix.

CONCLUSION

Odd-order Matrix Number Puzzles were studied here based on the concept of Arithmetic Sequences and the main goal is to place numbers in such a manner where the sum of each row, column, and principal diagonal is equal to a common sum. This research was done due to the necessity of having an approach to construct these puzzles that is easy to understand even by non-mathematically inclined people.

In order to achieve this aim, a line-based matrix construction technique was used that involved the use of the Straight-Line Left Angle Procedure (SLLAP) and the Straight-Line Right-Angle Procedure (SLRAP). With the aid of the SL-LAP and RAP process, an intermediate embossed matrix is generated wherein some arithmetic sequence numbers stay inside the matrix, while others go outside the matrix. This is where the IVPTCL method is employed to place the numbers outside the matrix inside the vacant spaces of the matrix.

The proposed procedure was demonstrated progressively using (3×3) , (5×5) , (7×7) , (9×9) , and (11×11) odd-order square matrices. The worked examples show that, after completing the **IVPTCL** Technique process, the resulting matrices satisfy the required condition that all row sums, column sums, and both principal diagonal sums are equal to the predetermined common summation value. For example, the demonstrated matrices based on consecutive arithmetic sequences (1,2,3...etc) produce common sums of 15 for the (3×3) matrix, 65 for the (5×5) matrix, 175 for the (7×7) matrix, and 369 for the (9×9) matrix. (Rathnayake, R. M. N. B. (2026))

An important contribution of the study is the transformation of an equal-sum matrix construction problem into a visual and procedural arrangement method. Rather than requiring the user to depend entirely on advanced algebraic formulations, the proposed approach provides a sequence of identifiable construction and value-placement steps. The methodology further establishes a foundation for translating these procedures into a Python-based computational algorithm, in which the matrix order (n), first term (a), and common difference (d) of an arithmetic sequence can be provided as input parameters. The proposed methodological approach includes a **mathematical proof based on the principle of mathematical induction** to establish the validity of the proposed construction method for **odd-order square matrices of arbitrary odd order**.

In summary, the results of this study show that the suggested procedure offers a systematic framework for the construction of odd-order equal-sum matrices using an arithmetic sequence. The validity of the suggested procedure for constructing the matrix is also verified mathematically by proving the procedure via mathematical induction. Mathematical induction is a tool that is applied in order to prove that the procedure of arranging the numbers can be systematically extended to higher order square matrices, where there will always be an equality of all sums of the rows, columns, and principal diagonal elements. The use of mathematical reasoning, inductive proofs, construction procedures, algorithmic rules, and computations makes this class of number-puzzle construction more comprehensible and accessible.

REFERENCES

- [1]. Rathnayake, R. M. N. B. (2026). Development and validation of a matrix diagonal summation method for odd-order arithmetic square matrices. *International Journal of Scientific Research and Management*, 14(9), 3111–3122. <https://doi.org/10.18535/ijssrm/v14i09.ec06>
- [2]. Block, S. S., & Tavares, S. A. (2009). *Before Sudoku: The world of magic squares*. Oxford University Press. <https://doi.org/10.1093/oso/9780195367904.001.0001>
- [3]. Chan, C.-Y. J., Mainkar, M. G., Narayan, S. K., & Webster, J. D. (2014). A construction of regular magic squares of odd order. *Linear Algebra and its Applications*, 457, 293–302. <https://doi.org/10.1016/j.laa.2014.05.032>
- [4]. Maji, A. K., Jana, S., & Pal, R. K. (2013). An algorithm for generating only desired permutations for solving Sudoku puzzle. *Procedia Technology*, 10, 392–399. <https://doi.org/10.1016/j.protcy.2013.12.375>
- [5]. Pamplona, J. V., Pinheiro, M. E., & Santos, L.-R. (2026). Constructing magic squares: An integer constraint satisfaction problem and a fast approach. *Computational and Applied Mathematics*, 45, Article 332. <https://doi.org/10.1007/s40314-026-03731-3>
- [6]. Sesiano, J. (2019). *Magic squares: Their history and construction from ancient times to AD 1600*. Springer. <https://doi.org/10.1007/978-3-030-17993-9>
- [7]. Soto, R., Crawford, B., Galleguillos, C., Monfroy, E., & Paredes, F. (2013). A hybrid AC3-tabu search algorithm for solving Sudoku puzzles. *Expert Systems with Applications*, 40(15), 5817–5821. <https://doi.org/10.1016/j.eswa.2013.05.019>