



Programming with MATLAB to color latin squares

A. Shokri¹ orcid.org/0000-0002-2079-4392

M. Golriz² orcid.org/0000-0002-3215-3015

M. Alaeiyan³ orcid.org/0000-0003-2185-5967

¹Islamic Azad University-South Tehran Branch, Dept. of Mathematics, Theran, Iran.

✉ st_a.shokri@azad.ac.ir

²Islamic Azad University-South Tehran Branch, Dept. of Mathematics, Theran, Iran.

✉ golriz@azad.ac.ir

³Iran University of Science and Technology, Dept. of Mathematics, Theran, Iran.

✉ alaeiyan@iust.ac.ir

Received: February 2020 | Accepted: May 2020

Abstract:

With a Matlab programming we will find the chromatic number for all Latin squares of order smaller than 7. Previously, a manual algorithm for coloring the Latin square was provided. This algorithm determined the chromatic number of some special classes of Latin squares such as Cyclic or Dihedral, so, we tried to speed up the process of this algorithm with a programming.

Keywords: Latin square; Transversal; Partial transversal; Chromatic number; Dihedral group.

MSC (2020): 05B15, 05C15, 68R10.

Cite this article as (IEEE citation style):

A. Shokri, M. Golriz, and M. Alaeiyan, "Programming with MATLAB to color latin squares", *Proyecciones (Antofagasta, On line)*, vol. 39, no. 4, pp. 933-943, Aug. 2020, doi: 10.22199/issn.0717-6279-2020-04-0058.



Article copyright: © 2020 A. Shokri, M. Golriz and M. Alaeiyan. This is an open access article distributed under the terms of the Creative Commons License, which permits unrestricted use and distribution provided the original author and source are credited.



1. Introduction

A *Latin square* of order n is an $n \times n$ array of symbols such that each symbol occurs once in each row and once in each column. Let (G, \circ) be a finite group of order n . A Cayley table for G is an $n \times n$ matrix, denoted by L_G , where the cell (i, j) contains the group element $g_i \circ g_j$, for some fixed enumeration $G = \{g_0, \dots, g_{n-1}\}$. It is easy to see that L_G is a Latin square. If G is a cyclic group, then L_G is called a circulant Latin square [1].

A partial transversal of L as a collection of cells which intersects each row, each column, and each symbol class at most once. A transversal of L is a partial transversal of size n .

Let L be a *Latin square* of order n , The chromatic number of L , denoted by $\chi(L)$, is the minimum number of partial transversal of L which together cover the cells of L .

All groups with transversal have chromatic number equal to their order and for groups of order n which have no transversal, the chromatic number can not be n or $n + 1$ [2].

Theorem 1.1. *We have*

1. $\chi(G) = |G|$ for every group G of odd order.
2. For every group G of order n , either $\chi(G) = n$ or $\chi(G) \geq n + 2$. [1]

To compute the chromatic number of Latin square, the author has written an algorithm and proved that the Cayley table of some groups of order n , which has no transversal, has a chromatic number equal to $n + 2$. [5]

The group of symmetries of an n -sided regular polygon for $n > 1$ with rotations and reflections is called Dihedral group and is denoted by D_n . The order of the Dihedral group is $2n$. D_n is a non-abelian group for $n > 2$ [4,6].

It has been shown before [2] that D_n group (for even n) has transversal. So the Latin square resulted from it's Cayley table is colored with $2n$ (=group order).

Also Coloring the Cayley table of D_n group for prime and odd, has been previously done by researchers. they look for partial transversals to use the minimum number of colors, and it has been shown, if $n > 3$ (n is prime) then $\chi(L_{D_n}) = 2n + 2$. [3]

Although the algorithm mentioned [5] was the reliable way to color Latin square obtained from group D_n for all n and other groups such as

$z_2 \times z_4$ and the Cyclic Latin square for even n , but for coloring Latin square of the high order, using this algorithm needs more accuracy and time.

So for accelerating the process of implementing the algorithm, we decided to write a suitable program for this algorithm and we chose matlab.

Matlab computing software is a powerful and intelligent computer algebraic system for performing numerical calculations. This programming language, is an interacting environment for numeral computing and programming. [2]

Its name is chosen from a combination of the words matrix and laboratory. This name indicates the matrix-based approach of the program in which even single numbers are considered as a 1×1 matrix. Also a string like "matrix laboratory" are saved as a matrix with one row and several columns. Its columns are as many as phrase characters.

Description of the coloring program

The reason for choosing matlab is that its inputs can be a matrix. Through Excel software, high-order matrices can be given as input.

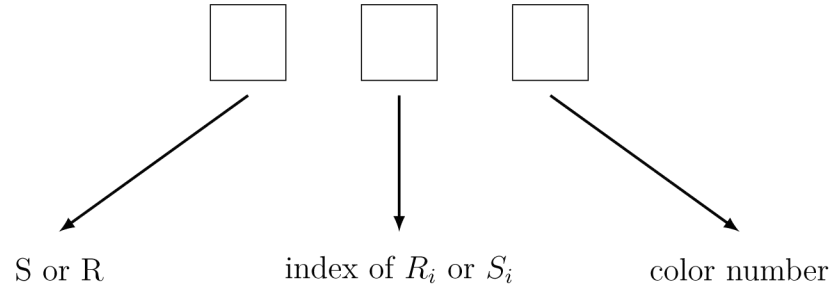
Its enough to consider the Cayley table of the group as a matrix .but the input data in matlab should be a matrix with numerical entries. In this case, we need to convert the Cayley table entries to numbers.

In D_n group we should consider the matrix entries as follows:

D_n group members are as R_i or S_i so we consider the first position on the left as the number 1 instead of R_i symbol and number 2 instead of S_i symbol.

In the second position numbers 0 to $n - 1$ instead of index of R_i or S_i .

In the third position, numbers 1 to $2n + 2$ instead of color number in coloring D_n .



We see that as the matrix order increases, the position of the color number and index may be a two-digit number or even more. For example, if we consider the Cayley table of D_3 group:

| | | | | | |
|-------|-------|-------|-------|-------|-------|
| R_0 | R_1 | R_2 | S_0 | S_1 | S_2 |
| R_1 | R_2 | R_0 | S_1 | S_2 | S_0 |
| R_2 | R_0 | R_1 | S_2 | S_0 | S_1 |
| S_0 | S_2 | S_1 | R_0 | R_2 | R_1 |
| S_1 | S_0 | S_2 | R_1 | R_0 | R_2 |
| S_2 | S_1 | S_0 | R_2 | R_1 | R_0 |

Table 1.1:

The input matrix will be as follows:

| | | | | | |
|----|----|----|----|----|----|
| 10 | 11 | 12 | 20 | 21 | 22 |
| 11 | 12 | 10 | 21 | 22 | 20 |
| 12 | 10 | 11 | 22 | 20 | 21 |
| 20 | 22 | 21 | 10 | 12 | 11 |
| 21 | 20 | 22 | 11 | 10 | 12 |
| 22 | 21 | 20 | 12 | 11 | 10 |

Table 1.2:

In this way, the Cayley table of D_3 group is colored as follows:

| | | | | | |
|-----|-----|-----|-----|-----|-----|
| 101 | 114 | 126 | 202 | 218 | 225 |
| 112 | 121 | 107 | 215 | 226 | 203 |
| 123 | 108 | 115 | 221 | 204 | 212 |
| 207 | 222 | 214 | 103 | 125 | 111 |
| 216 | 205 | 223 | 117 | 102 | 124 |
| 224 | 217 | 201 | 128 | 113 | 106 |

Table 1.3:

For coloring 3×3 matrix which is generally shown as follows:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

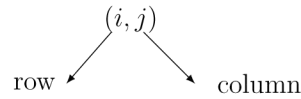
We want to color the matrix with 3 colors, each colored cell is shown with a number as a color number mark.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$
$$\begin{vmatrix} 1 & a_{12} & a_{13} \\ a_{21} & 1 & a_{23} \\ a_{31} & a_{32} & 1 \end{vmatrix}$$
$$\begin{vmatrix} 1 & 2 & a_{13} \\ a_{21} & 1 & 2 \\ 2 & a_{32} & 1 \end{vmatrix}$$
$$\begin{vmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \\ 2 & 3 & 1 \end{vmatrix}$$

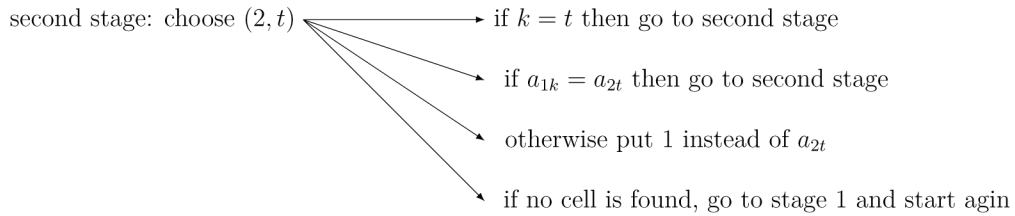
At high order this color allocation may not be possible and the number of processes and colors may increase.

coloring flowchart

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}$$



coloring flowchart in general matrix first stage: choose $(1, K)$ - - - put 1



By completing step n , we will generally have a partial transversal.

According to the above mentioned flowchart, The number of modes for selecting cells in general

First stage $n!$

Second stage $(n - 2)!$

Third stage $(n - 4)!$

The number of modes for selecting cells for D_n is $2n!$. In other words for example D_{15} for selecting cells in first stage, there are 30 modes. To explain the program process, first we consider this Latin squares of order 4 or 4×4 matrix:

| | | | |
|---|---|---|---|
| 4 | 1 | 2 | 3 |
| 1 | 4 | 3 | 2 |
| 2 | 3 | 4 | 1 |
| 3 | 2 | 1 | 4 |

Table 1.4:

First we color first line with 4 colors then we start from left column and we color each cell from top to bottom according to the coloring conditions and in order of color numbers.

| | | | |
|---|---|---|---|
| 4 | 1 | 2 | 3 |
| 1 | 4 | 3 | 2 |
| 2 | 3 | 4 | 1 |
| 3 | 2 | 1 | 4 |

Table 1.5:

If we color the columns one by one from left to right:

| | | | |
|---|---|---|---|
| 4 | 1 | 2 | 3 |
| 1 | 4 | 3 | 2 |
| 2 | 3 | 4 | 1 |
| 3 | 2 | 1 | 4 |

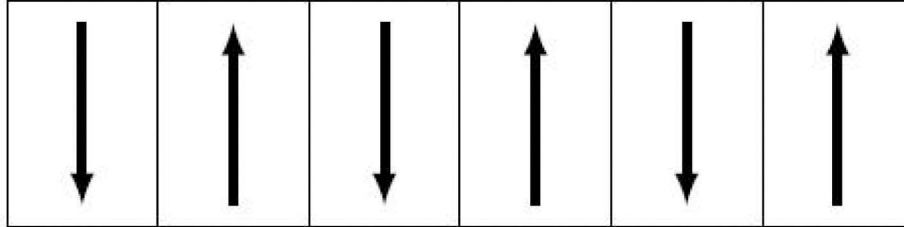
Table 1.6:

In fact the colored matrix has been colored in the standard way. But coloring isn't always that simple we consider this example:

Initially, the program starts coloring by default, as in the first case. But this matrix is not colored with above mentioned method. In second method we try to increase the scatter. Like previous mode we color the first row with 6 colors (= Matrix order). at this stage unlike the previous method, we assign each color from the left column from top to bottom and next column from bottom to the top.

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 3 | 1 | 5 | 6 | 4 |
| 3 | 1 | 2 | 6 | 4 | 5 |
| 4 | 6 | 5 | 1 | 3 | 2 |
| 5 | 4 | 6 | 2 | 1 | 3 |
| 6 | 5 | 4 | 3 | 2 | 1 |

Table 1.7:



Then we go to the next color. In fact here we use partial transversal. Another difference in this coloring mode is the uniform distribution of scatters of the cells. It means in colors with odd numbers we move from the first left column from top to bottom and we go forward in a spiral. In colors with even numbers we move from the last right column from bottom to the top and go forward in a spiral.

| | |
|---|--|
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 3 | 1 | 5 | 6 | 4 |
| 3 | 1 | 2 | 6 | 4 | 5 |
| 4 | 6 | 5 | 1 | 3 | 2 |
| 5 | 4 | 6 | 2 | 1 | 3 |
| 6 | 5 | 4 | 3 | 2 | 1 |

Table 1.8:

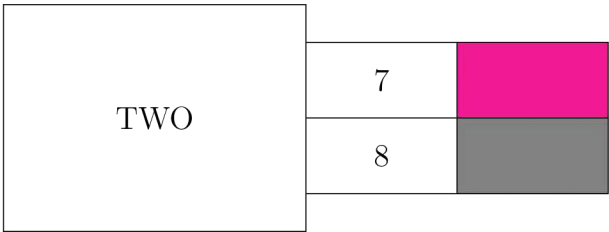
| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 3 | 1 | 5 | 6 | 4 |
| 3 | 1 | 2 | 6 | 4 | 5 |
| 4 | 6 | 5 | 1 | 3 | 2 |
| 5 | 4 | 6 | 2 | 1 | 3 |
| 6 | 5 | 4 | 3 | 2 | 1 |

Table 1.9:

When finished we should check that in each column and row we should’nt have more than two uncolored cells this checks the cells with this condition:

```
[ip,jp]=find(G==0);
n=numel(G(:,1));
for kind=1:n
    FFi=find(ip==kind);
    [ui,uj]=size(FFi);
    Ui(kind)=ui;
    FFj=find(jp==kind);
    [ui,uj]=size(FFj);
    Uj(kind)=ui;
end
UU=[Ui,Uj];
dddd=find(UU>=3);
```

Which is named with two function in this program



| | | | | | |
|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 |
| 2 | 3 | 1 | 5 | 6 | 4 |
| 3 | 1 | 2 | 6 | 4 | 5 |
| 4 | 6 | 5 | 1 | 3 | 2 |
| 5 | 4 | 6 | 2 | 1 | 3 |
| 6 | 5 | 4 | 3 | 2 | 1 |

Table 1.10:

| | | | | | |
|----|----|----|----|----|----|
| 11 | 22 | 33 | 44 | 55 | 66 |
| 23 | 34 | 15 | 52 | 61 | 48 |
| 35 | 16 | 21 | 68 | 42 | 53 |
| 47 | 65 | 58 | 13 | 36 | 24 |
| 56 | 43 | 62 | 25 | 14 | 31 |
| 64 | 51 | 46 | 37 | 28 | 12 |

Table 1.11:

So this program colors all Latin squares up to order 6. All Latin squares are colored, whether they have transversal or partial transversal.

We hope that in the future researchers will be Upgrade this programming.

Acknowledgements

I would like to thank Mr. Saeed Mozaffari for his help in writing the coloring program under MATLAB programming.

References

- [1] N. Besharati, L. Goddyn, E. S. Mahmoodian, and M. Mortezaeefar, "On the chromatic number of latin square number of latin square graphs", *Discrete mathematics*, vol. 339, no. 11, pp. 2613-2619, Nov. 2016, doi: 10.1016/j.disc.2016.04.025

- [2] T. Mora, Ed., *Applied algebra, algebraic algorithms and error-correcting codes*. Berlin: Springer, 1989, doi: 10.1007/3-540-51083-4
- [3] M. Mortezaeefar, "Colorings of latin square graphs and block designs", Master's thesis, Sharif University of Technology, 2009.
- [4] K. Shokri, "On the latin square of groups and their coloring", Master's thesis, Sharif University of Technology, 2015.
- [5] A. Shokri and M. Golriz, "A new algorithmic method to compute the chromatic number of dihedral group", *in press*.
- [6] I. Wanless, "Transversals in latin squares: a survey," in *Surveys in Combinatorics 2011*, R. Chapman, Ed. Cambridge: Cambridge University Press, 2011, pp. 403–437, doi: 10.1017/CBO9781139004114.010