Algorithms in Finding All the 4 by 4 Magic Matrices

Zheng Chen  
Natural Sciences Department  
Southern University at New Orleans  
New Orleans, LA USA 70126  
zchen@suno.edu

Abstract—In this paper, based on the Gauss-Jordan elimination to solve linear systems, the algorithms for finding all the magic matrices of size 4 by 4 are developed and discussed, and the efficiency of the algorithms is tested after complementing the algorithms in MATLAB. The total magic matrices of 4 by 4 can be found with the program; for completeness and explaining the algorithms, the case of 3 by 3 is also included.

Keywords—Gauss-Jordan elimination; Echelon form; RREF; magic matrices; algorithm

I. INTRODUCTION

A n by n magic square is a square array of the numbers from 1 to \( n^2 \) in which each row, each column, and the two main diagonals sum to the same number. The first clear magic square, denoted as M, was described in the Ta Tai Li-Chi compiled in the first century A.D. from older sources and the invention of the magic square is attributed to the Chinese by Schuyler Cammann [2]. Examples were also found in Arab and Indian cultures around the 7th century A.D. For a long time, magic squares have been investigated by many mathematicians from different approaches. The basic interest is to find all the magic matrices of different sizes. Here, we like to call a magic matrix as a magic square in order to differentiate from the magic cubes in a 3-D spaces. For more reference, see [4], [5], [5] and [10]. In this paper, we will propose a very efficient algorithm to find all magic matrices of 4 by 4, the main tool we use is the Gauss-Jordan elimination to solve a linear system; to complement the algorithm, we use MATLAB, a powerful software for scientific computing. MATLAB has been extensively used in teaching in colleges or even high schools, research and industry.

Magic squares, or magic matrices provide us good topics for undergraduate research, through this paper, we hope to communicate with mathematics professors in advising undergraduate students to do research as well as undergraduate students.

To make it clear, the following matrix of 3 by 3

\[
\begin{bmatrix}
  x_1 & x_2 & x_3 \\
  x_4 & x_5 & x_6 \\
  x_7 & x_8 & x_9
\end{bmatrix}
\]

is a magic square if all the elements \( x_1, x_2, ..., x_9 \) are a permutation of integers from 1 to 9, and all the sums of each row, each column, each diagonal are equal.

We know the sum of integers from 1 to n is \( n(1+n)/2 \); it follows that the sum of integers from 1 to 9 is 45, and then the sum of each row in a magic matrix of 3 by 3 is 15. This number 15 is called the magic number of the 3x3 square.

For a magic matrix above, the conditions can be written as a linear system: \( Ax = b \), where

\[
A = \begin{bmatrix}
  1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
  1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
  0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
  1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
  0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0
\end{bmatrix}
\]

\[
X = [x_1, x_2, ..., x_9]'
\]
\( \mathbf{b} = [15, \ldots, 15]' \).

We will mainly work with the augmented matrix \([A, \mathbf{b}]\), which is of size 8 by 10. The system \( Ax = b \) is overdetermined: it has 9 unknowns and 8 equations; besides, we look for integers solutions, also a permutation of integers from 1 to 9.

Similarly we have the definition of a magic matrix of 4 by 4, which has its magic number 34, and the linear system \( Ax = b \), where \( A \) has a size 10 by 16, \( X \) and \( b \) have a same size 16 by 1, and each element in \( b \) is 34.

### II. MAGIC MATRICES OF RANK THREE

For completeness, we like to cover the algorithm to get all the magic matrices of 3 by 3; this case is easy to understand and can be also intuitive to the case of 4 by 4.

For a magic matrix of size 3 by 3, there is a simple fact:
\( x_5 = 5 \).

Actually, taking out two diagonals, the 2\(^{\text{nd}}\) row and 2\(^{\text{nd}}\) column, summing up the all elements in these 4 vectors, in this sum, all elements are counted once except \( x_5 \), which is counted 4 times. On the other hand, we know the total sum of a magic matrix is 45 and each vector has a sum 15, therefore, we have \( 45 + 3x_5 = 4 \times 15 \), which gives \( x_5 = 5 \).

With this fact, the system \( Ax = b \) can be reduced to a new system with 8 unknowns. We need to solve \( Cx = d \), where

\[
C = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
\end{bmatrix}
\]

\[
\mathbf{x} = [x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9]',
\]

\[
d = [15 10 15 10 15 10 10]'\]
\[
\begin{align*}
X_1 &= 10 - X_9 \\
X_2 &= 10 - X_8 \\
X_3 &= -5 + X_8 + X_9 \\
X_4 &= -10 + X_8 + 2X_9 \\
X_5 &= 20 - X_8 - 2X_9 \\
X_6 &= 15 - X_8 - X_9
\end{align*}
\]

Based on the structure of this solution space, we have the following algorithm which can implement in MATLAB without any difficulty.

1. For all \(X_8\) and \(X_9\) taking all values from 1 to 9
2. Using the above equations the calculate the values of \(X_1, X_2, X_3, X_4, X_6\) and \(X_7\).
3. Check whether the solution vector is a permutation of integers 1 to 9.

Using this algorithm, we can do the programming in MATLAB pretty easy and run it to get the solutions in 1 second, it gives 8 different solutions, and they are essentially the reflection or rotation of matrix M in introductory section I.

III. MAGIC MATRICES OF RANK FOUR

As in the case of 3 by 3, we are dealing with the magic matrix of 4 by 4 using the Reduced Row-Echelon form. In this case, the conditions can be written as a linear system: \(AX=b\), where

\[
\begin{align*}
A &= [1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1] \\
B &= [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
\end{align*}
\]

From the Reduced Row-Echelon Form resulted from \texttt{RREF}(B) in MATLAB, we know the structure of the solution space; it tells that the rank of this form is 9, it follows that there are 7 independent unknowns; 9 unknowns can be expressed by other 7 unknowns.

\[
\begin{align*}
X_1 &= x_8 + x_{12} + x_{14} + x_{15} + x_{16} - 34 \\
X_2 &= x_8 - x_{10} + x_{11} + x_{12} + x_{15} + 2x_{16} - 34 \\
X_3 &= -x_8 + x_{10} - x_{11} - x_{12} - x_{14} - 2x_{15} - 2x_{16} + 68 \\
X_4 &= -x_8 - x_{12} - x_{16} + 34 \\
X_5 &= -x_8 + x_{10} + x_{11} \\
X_6 &= -x_8 - x_{11} - x_{12} - x_{14} - x_{15} - 2x_{16} + 68 \\
X_7 &= x_8 - x_{10} + x_{12} + x_{14} + x_{15} + 2x_{16} - 34 \\
X_8 &= x_{10} + x_{11} + x_{12} = 34 \\
X_9 &= x_{13} + x_{14} + x_{15} + x_{16} = 34
\end{align*}
\]

IV. DEVELOPING OF TWO ALGORITHMS

Algorithm I

Step 1. for all permutation \(x_8, x_{10}, x_{11}\) and \(x_{12}\) from 1 to 16 with a sum 34;

Step 2. for all permutation \(x_{13}, x_{14}, x_{15}\) and \(x_{16}\) from 1 to 16 with a sum 34;

Step 3. for \(X_8\) taking values from 1 to 16 and different from the above values.;
Step 4. using the above equations to calculate the values 
\( x_1, x_2, x_3, x_4, x_5, x_6 \) and \( x_7 \);
Step 5. check whether \( x_1, x_2, \ldots, x_{15} \) and \( x_{16} \) is a permutation of 1, 2, ..., 16.

In steps, all the vectors \([x_1, x_2, \ldots, x_{15}, x_{16}]\)
which are permutations will give all the magic matrices of 4 by 4.

In the following, we will make an alteration in the algorithm I, we note that \([x_9, x_{10}, x_{11}, x_{12}]\) and
\([x_{13}, x_{14}, x_{15}, x_{16}]\) satisfy the same conditions, therefore, we may take use of the vectors in step 1 to get the vector in step 2 and we have the following:

Algorithm II
Step 1. take all all permutation \( x_9, x_{10}, x_{11} \) and \( x_{12} \) from 1 to 16 with a sum 34 and save them in a matrix as rows;
Step 2. take any two disjoint rows in above matrix;
Step 3. for \( x_8 \) taking values from 1 to 16 and different from the above values;
Step 4. using the above equations to calculate the values 
\( x_1, x_2, x_3, x_4, x_5, x_6 \) and \( x_7 \);
Step 5. check whether \( x_1, x_2, \ldots, x_{15} \) and \( x_{16} \) is a permutation of 1, 2, ..., 16.

In algorithm II, with all the possible permutations for
\([x_9, x_{10}, x_{11}, x_{12}]\) and \([x_{13}, x_{14}, x_{15}, x_{16}]\),
we actually have only one independent variable \( x_8 \), which will save a lot of time and make the computation much more efficient!

V. IMPLEMENTATION AND THE SCRIPT FILE

We develop the Algorithms I and II and implement them in MATLAB R2007b, a. Besides, all experiments are implemented under the Windows 8.1 Operating System, 6 GB RAM and Intel® Core™ 2 i5-4200U CPU. When using
algorithm I, it takes about 19.02 minutes to get all the 7040 unique magic squares of 4 by 4; while using algorithm II, it only takes 5.91 minutes; also comparing with 45 minutes in paper [6], it concludes that the algorithm II is a very efficient one. For the convenience of readers, the script file in Matlab is included in the following.

% this program gives all the magic matrices of 4 by 4 as well as the total number of magic matrices..
% with the matrices A, b, B, RD in section III.
% SSS is a magic matrix and iter is the total number of magic matrices of 4 by 4.
tic
M=[];
for a=1:16
    for b=1:16
        d=34-a-b-c;
        if 0 <d & d<17
            x=[a,b,c,d];
            if length(unique(x))==4
                M=[M;x];
            end
        end
    end
end
m=size(M); n=m(1,1); iter=0;
for k=1:n
    for p=1:n
        C=[M(k,:), M(p,:)];
        if length(unique(C))==8
            for a8=1:16
                if sum(C==a8)==0
                    D=[a8, C]';
                    E=unique(D);
                end
            end
        end
    end
end
4
if length(E)==9
    S7= RD(1:7, 17)-RD(1:7, 8:16)*D;
    % the first 7 values in a solution
    SS=S7; SS=[SS; D];
    aa=sort(SS);
    if aa==[1:16]
        iter=iter+1;
        SSS = reshape(SS,[4,4]);
        % a magic matrix
    end
end
end
end
end
toc

ACKNOWLEDGMENT

This research is partially supported during 2019 summer by the Grant: 4D9920-00 R & D Coord STEAM Grant.

REFERENCES

[1] BIANCA EDWARDS and JIM HARTMAN: Powers of magic matrices, The Mathematical Gazette, Vol. 95, No. 533 (July 2011), pp. 284-292.
[2] Schuyler Cammann, The evolution of magic squares in China, Journal of the Oriental Society, 80, No. 2 (Apr. - Jun. 1960), pp. 116-124.
[3] Lay, David C. (2011): Linear Algebra and its Applications, Pearson, 4th edition.
[4] C. R. Johnson, A matrix theoretic construction of magic squares, The American Mathematical Monthly, 79, No. 9 (Nov. 1972), pp. 1004-11006.
[5] J. L. Chabert, Magic squares, A history of algorithms: from the pebble to the microchip, Springer-Verlag, New York, 1999, pp. 49
[6] Changyu Liu, Tiezhu Zhao and Bin Lu: An Efficient Algorithm for Constructing all Magic Squares of Order Four, International Journal of Control and Automation Vol.8, No.11 (2015), pp.235-244 http://dx.doi.org/10.14257/ijca.2015.8.11.23
[7] http://mathworld.wolfram.com/MagicSquare.html
[8] https://pdfs.semanticscholar.org/dcdb/c3d9e7e31b4d9f9c1a94d4678258f077917.pdf
[9] https://www.mathworks.com/moler/exm/chapters.html
[10] Swetz, Frank J. (2008). The Legacy of the Luoshu (2nd Rev ed.). A. K. Peters / CRC Press. ISBN 978-1-56881-427-8.