

GENERATING AND VISUALIZING LATIN
CUBE AND SUDOKU CUBE

NG E LUEN

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE WITH
HONORS

PERPUSTAKAAN
UNIVERSITI MALAYSIA SABAH

MATHEMATICS WITH COMPUTER GRAPHICS PROGRAM
SCHOOL OF SCIENCE AND TECHNOLOGY
UNIVERSITI MALAYSIA SABAH

April 2007



UMS
UNIVERSITI MALAYSIA SABAH

BORANG PENGESAHAN STATUS TESIS@

JUDUL: Generating and Visualizing Latin Cube and Sudoku CubeIjazah: Degree of Bachelor of Science. (Sarjana Muda SainsSESI PENGAJIAN: Tahun 3.Saya NG E LUEN

(HURUF BESAR)

mengaku membenarkan tesis (LPS/Sarjana/Doktor Falsafah)* ini disimpan di Perpustakaan Universiti Malaysia Sabah dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Sabah.
2. Perpustakaan Universiti Malaysia Sabah dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (/)

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

luen

(TANDATANGAN PENULIS)

Disahkan oleh

dy ole

(TANDATANGAN PUSTAKAWAN)

Alamat Tetap: 130-A, Kampung Cina,
Parit 21, 36400Wutan Melintang, Perak.

Nama Penyelia

Tarikh: 19/04/2007.

Tarikh: _____

CATATAN: * Potong yang tidak berkenaan.

** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh tesis ini perlu dikelaskan sebagai SULIT dan TERHAD.

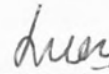
@ Tesis dimaksudkan sebagai tesis bagi Ijazah Doktor Falsafah dan Sarjana secara penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (LPSM).



DECLARATION

I hereby declare that this dissertation is my own work except for the summaries and quotations whereby each of its origin and source has been duly acknowledged.

12 March 2007



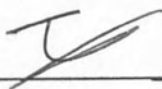
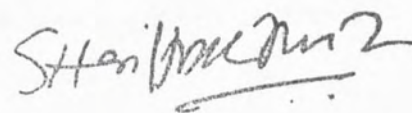
NG E LUEN

HS2004 – 2252



AUTHENTICATION

Signature

1. SUPERVISOR**(Mr. Tiong Kung Ming)**
_____**2. EXAMINER 1****(Mr. Rajasegeran Ramasamy)**
_____**3. DEAN****(SUPT./KS. Assoc. Prof Dr. Shariff A.K. Omang)**
_____

ACKNOWLEDGEMENT

First, I would like to thank my supervisor, Mr. Victor Tiong Kung Ming for his guidance and advices in helping me to eliminate the problems throughout the writing process for this dissertation. Without his help, this dissertation will not be completed successfully.

Several of my friends have provided some valuable ideas, which are very helpful for this dissertation. Here, I specially thank to Mr. Lim Keng Poh. He always tries to help me to solve the problems when I am facing the programming problem. By his help, the programs can be run properly and effectively. I would also like to thank Mr, Ganesan Armugam, who has provided some idea of Latin Square and SuDoku for my dissertation. He also gave a lot of advice in playing these games, which is needed to complete the methodology for this dissertation.

Finally, I would like to thank my family, which give the greatest support throughout the writing process. And also, I would like to thank anyone, who has contributed in this dissertation. I am sorry if I did not mention your name here but I will always appreciate your help in this dissertation.



ABSTRACT

Latin Square and SuDoku is $n \times n$ array and can extend to three dimensions which can form a cube. This research is to determine whether different layers of Latin Square or SuDoku can form a cube or not, which is called Latin Cube and SuDoku Cube respectively. This research also is to investigate whether a Latin Cube or SuDoku Cube is unique or not. Row interchanging is used to find the matching layer for a Latin Square to form a cube. Method 1 and Method 2 are used to find the matching layer for a SuDoku to form a cube. The basic idea of Method 1 and Method 2 is from row interchanging and column interchanging. Two programs are written in C++ computer language by using these methods, for Latin Cube and SuDoku Cube respectively. These programs can find the matching layer for a Latin Square and SuDoku, and save the outputs in VRML file type (.wrl). These outputs, or cube, can be viewed in a 3D environment by using VRML. The cubes can be viewed and studied in detail using VRML.



PENGHASILAN DAN PEMBAYANGAN KUBUS LATIN DAN KUBUS SUDOKU

ABSTRAK

Segiempat Latin dan SuDoku adalah tersusun dalam dua dimensi, dan boleh dilanjutkan ke tiga dimensi untuk membentuk satu kubus. Kajian ini adalah untuk menentukan samada segiempat Latin atau SuDoku yang tidak sama, boleh membentuk satu kubus atau tidak. Kubus ini masing-masing dipanggil Kubus Latin dan Kubus SuDoku. Kajian ini juga untuk menyelidik samada satu Kubus Latin atau Kubus SuDoku adalah unik atau tidak. Teknik penukaran baris digunakan untuk mencari lapisan yang sesuai bagi sesuatu Segiempat Latin supaya dapat membentuk satu kubus. Cara 1 dan Cara 2 digunakan untuk mencari lapisan yang sesuai bagi sesuatu SuDoku supaya dapat membentuk satu kubus. Idea asas bagi Cara 1 dan Cara 2 adalah dari teknik penukaran baris dan lajur. Dua program telah ditulis dalam satu bahasa komputer, iaitu C++, dengan menggunakan cara-cara ini dan masing-masing untuk mencari Kubus Latin dan Kubus SuDoku. Program-program ini boleh mencari lapisan yang sesuai untuk sesuatu Segiempat Latin dan SuDoku, dan outputnya akan disimpan sebagai jenis fail untuk VRML (.wrl). Output-output atau kubus-kubus ini boleh dipandang dalam satu keadaan 3D dengan menggunakan VRML. Kubus-kubus dapat dipandang dan dikaji dengan lebih jelas di dalam VRML.



CONTENTS

Page Number

DECLARATION	ii
AUTHENTICATION	iii
ACKNOWLEDGEMENT	iv
ABSTRACT	v
ABSTRAK	vi
CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER 1 INTRODUCTION	
1.1 BACKGROUND	1
1.2 OBJECTIVE OF RESEARCH	4
1.3 SCOPE OF RESEARCH	4
1.4 HYPOTHESIS	5
CHAPTER 2 LITERATURE REVIEW	
2.1 LATIN SQUARE	6
2.2 SUDOKU	9
2.2.1 Brief History of SuDoku	9
2.2.2 Strategies to Solve SuDoku	10
2.2.3 Total Number of Possible SuDoku Grid	11
2.2.4 Dion Cube	11
CHAPTER 3 METHODOLOGY	
3.1 BACKGROUND	13
3.2 GENERATING A LATIN CUBE	13
3.3 GENERATING A SUDOKU CUBE	17
3.3.1 Method 1($m \times m$ Region Input)	17
3.3.2 Method 2(Initial Layer Input)	25
3.4 INVESTIGATION ON LATIN CUBE AND SUDOKU CUBE	30
3.4.1 Latin Cube	30
3.4.2 SuDoku Cube	31



3.5	VERIFICATION OF THE LATIN CUBE AND SUDOKU CUBE	32
3.6	ALGORITHM FOR LATIN CUBE AND SUDOKU CUBE	33
CHAPTER 4 RESULTS AND DISCUSSION		
4.1	RESULT	37
4.1.1	Output for Latin Cube	37
4.1.2	Output for SuDoku Cube	47
4.2	DISCUSSION	60
4.2.1	Limitations of the Programs	60
4.2.2	Limitations of VRML	64
CHAPTER 5 CONCLUSION AND SUGGESTIONS		
5.1	CONCLUSION	67
5.2	SUGGESTIONS FOR FURTHER RESEARCH	69
	REFERENCES	70



LIST OF TABLES

No. of Table		Page
2.1	The number of reduced Latin Square with different n	8
2.2	Total number of Latin Square	8



LIST OF FIGURES

No. of Figure	Page
1.1 Example of a Latin Square of order 3	1
1.2 Latin Square with different entries	1
1.3 An example of 9×9 SuDoku	2
1.4 A Latin Cube with $n = 4$	3
1.5 An example of SuDoku Cube with $m = 2$ and $n = 4$	3
2.1 An example of reduced Latin Square	6
2.2 An example of not a reduced Latin Square	7
2.3 An example of Dion Cube	12
3.1 4×4 Latin Square	14
3.2 From first layer to second layer	14
3.3 From second layer to third layer	15
3.4 From third layer to fourth layer	15
3.5 Latin Square in sequence	15
3.6 Row interchanging for order n Latin Square	16
3.7 Obtain different layers by using column interchanging	16
3.8 A SuDoku grid that will be used in explanation	17
3.9 Initial input	18
3.10 The way to obtain second and third regions	18
3.11 The way to obtain fourth region and seventh region	19
3.12 The way to obtain fifth and sixth regions	19
3.13 The way to obtain eighth and ninth regions	19
3.14 A Complete SuDoku grid	20
3.15 The way rows and columns interchange	21
3.16 From initial layer to second layer	21
3.17 From second layer to third layer	22
3.18 Obtaining the fourth, fifth and sixth layers from the initial layer	23
3.19 Obtaining the seventh, eighth and ninth layers from the fourth layer	24
3.20 An example for Method 2	25
3.21 Obtaining the second and third layers	26
3.22 Obtaining the fourth layer	27



3.23	Obtaining the fifth and sixth layers	28
3.24	Obtaining the seventh, eighth and ninth layers	29
3.25	Four Latin Squares obtained by using row interchanging	31
3.26	Four Latin Squares obtained by using column interchanging	31
3.27	First column from layer 1 to 9	33
3.28	Implementation of obtaining the Latin Cube	35
3.29	Implementation of obtaining SuDoku Cube	36
4.1	An example output of the program (layer 1 and 2)	38
4.2	An example output of the program (layer 3 and 4)	38
4.3	Viewing Latin Cube in VRML (from front or the original position)	39
4.4	Viewing Latin Cube in VRML from the top	40
4.5	Viewing the Latin Cube in VRML from the left side	41
4.6	Fit the cube (from Figure 4.5) to the window	41
4.7	Viewed the cube layer by layer (after fit to window)	42
4.8	Example output of a cube in order 9 (data input)	43
4.9	Example output of a cube in order 9 (layer1 and layer 6)	44
4.10	Example output of a cube in order 9 (layer 7 to layer 9)	45
4.11	Viewing the cube from the top and the left side	46
4.12	Viewing the cube layer by layer	47
4.13	Output of Option 1 (layer 1 and 2)	48
4.14	Output of Option 1 (layer 3 and 4)	49
4.15	Viewing SuDoku Cube in VRML (in original position)	50
4.16	Viewing SuDoku Cube form the top (fit to the window)	50
4.17	Viewing SuDoku Cube from the side (fit to the window)	51
4.18	View the SuDoku Cube layer by layer	52
4.19	Output of Option 1 with the size of region is 3 (input data)	52
4.20	Output of Option 1 with the size of region is 3 (layer 1 to layer 6)	53
4.21	Output of Option 1 with the size of region is 3 (layer 7 to layer 9)	54
4.22	Viewing SuDoku Cube in VRML from the back and top left corner	55
4.23	Viewing the SuDoku Cube layer by layer	56
4.24	Output of Option 2 (input data)	57
4.25	Output of Option 2 (layer 1 to layer 2)	57
4.26	Viewing SuDoku Cube in VRML from top right corner	58



4.27	Viewing SuDoku Cube in VRML from back top left corner	59
4.28	Viewing SuDoku Cube layer by layer	59
4.29	The error when the size is out of range	61
4.30	The error when the size is a non-numeric input	62
4.31	An example of a data which is entered wrongly	63
4.32	An order 4 Latin Cube	65
4.33	An order 15 Latin Cube	65



CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

A Latin Square in the order of n is an $n \times n$ array where the entries are from a set of n distinct symbols. These symbols are arranged in such a way that each symbol occurs exactly once in each row and in each column (Crownover & Thibault, 2005). Figure 1.1 and Figure 1.2 show examples of Latin Square in order 3 and order 4.

1	2	3
2	3	1
3	1	2

Figure 1.1 Example of a Latin Square of order 3

1	2	3	4
4	1	2	3
3	4	1	2
2	3	4	1

a	b	c	d
d	a	b	c
c	d	a	b
b	c	d	a

Figure 1.2 Latin Square order 4 with different entries

SuDoku is an $n \times n$ array where n is equal to m^2 , m is natural number. We can simplify it as a mathematical model:



$$n = m^2, m = 1, 2, 3 \dots \text{ or } m \in \mathbb{N}$$

The entries of a SuDoku is from a set of m^2 distinct symbols where are arranged in such way that every symbol occurs exactly once in each column, each row and each $m \times m$ region. These rules are similar to Latin Squares rules but there exists one extra rule which states that the symbols in an $m \times m$ region also must occur not more than once.

Latin Square and SuDoku both are $n \times n$ array. But the size of SuDoku is specified as $n = m^2$ which m is a positive integer. Therefore, we can conclude that SuDoku is subset of Latin Square. With the additional rule of SuDoku, it can be said that SuDoku is a special case of Latin Square (Weisstein, 2006). Any solution to a SuDoku puzzle is a Latin Square (Wikipedia, 2006). Figure 1.3 shows an example of 9×9 SuDoku.

5	6	9	3	1	7	2	4	8
8	1	4	2	9	6	3	5	7
3	7	2	5	8	4	1	9	6
6	2	1	9	4	5	8	7	3
7	8	5	1	3	2	9	6	4
9	4	3	6	7	8	5	1	2
2	3	7	4	5	9	6	8	1
4	9	6	8	2	1	7	3	5
1	5	8	7	6	3	4	2	9

Figure 1.3 An example of 9×9 SuDoku

From an order n Latin Square, if the rows for that Latin Square are rearranged, a different Latin Square will exist. It will give the same result if the columns are rearranged. So, if the procedure is repeated with $n-1$ times, $n-1$ new Latin Square will be obtained. Consequently, these Latin Squares will form a cube (call it Latin Cube) if

they are combined together. This is the main idea of the method that will be discussed in this research. Figure 1.4 shows a Latin Cube.

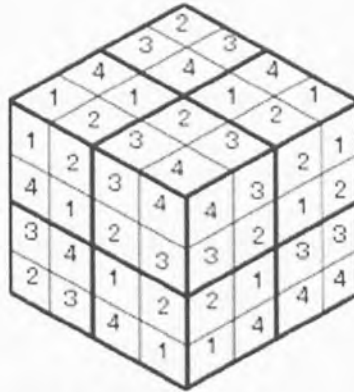


Figure 1.4 A Latin Cube with $n = 4$

The same idea will be applied in SuDoku. But, SuDoku consist an extra rule which leads to a higher complexity to find the matching layers. The interchanging of rows and columns will be applied at the same time to a SuDoku. The purpose is to eliminate the possibility of clashing among the elements in an $m \times m$ region. Figure 1.5 shows an example of $4 \times 4 \times 4$ SuDoku Cube.

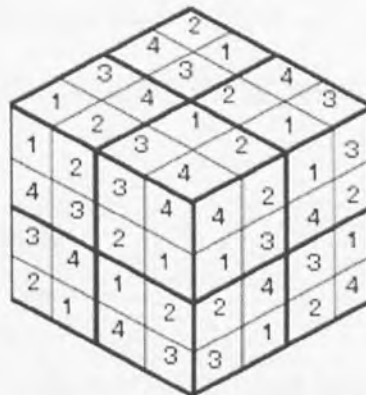


Figure 1.5 An example of SuDoku Cube with $m = 2$ and $n = 4$

From Figure 1.3 and Figure 1.4, the Latin square and SuDoku seem just lying on the surface of cube. In fact, it does not only occur on the surface of the cube. If the cube is separated into layer, every layer, no matter vertical or horizontal, a Latin Square or SuDoku will be obtained respectively. It means that the cube is formed by combining n layers of Latin Square or SuDoku where $n \in N$. It is hard to imagine or view the Latin Cube and SuDoku Cube in 2D form. The Latin Cube and SuDoku Cube will be viewed in a 3D environment, in Chapter 4.

1.2 OBJECTIVES OF RESEARCH

This research has four main objectives:

- i. To determine whether different layers of Latin Square can form a Latin Cube that fulfils Latin Square rules.
- ii. To determine whether different layers of SuDoku can form a SuDoku cube that fulfils SuDoku rules.
- iii. To create a program that can produce a Latin Cube and SuDoku Cube respectively, by obtaining the first layer.
- iv. To investigate whether a Latin Cube or SuDoku Cube is unique or not.

1.3 SCOPE OF RESEARCH

This research just covers the method of finding matching layers to form a Latin Cube or SuDoku Cube. This research will not touch on methods that are used to solve a Latin Square or a SuDoku. The methods that are discussed in this research also can not be used to solve any Latin Cube or SuDoku Cube. It is only about how to generate



and visualize the Latin Cube and SuDoku Cube. The size of Latin Cube can up to $80 \times 80 \times 80$. The number larger than that is not allowed because of the limitation of compiler, processor and system memory. The size of SuDoku Cube can up to $64 \times 64 \times 64$ ($m = 8$). The size $81 \times 81 \times 81$ ($m = 9$) SuDoku Cube is not allowed because of the same reason.

1.4 HYPOTHESIS

The methods discussed in this research will find the matching layers for a Latin Square or SuDoku to form a Latin Cube and SuDoku Cube. The set of matching layers will not be unique because different Latin Cube and SuDoku Cube can be formed from an initial layer.



CHAPTER 2

LITERATURE REVIEW

2.1 LATIN SQUARE

In 1783, a great mathematician, Leonhard Euler introduced Latin Square as a "nouveau espece de carres magiques", means a new kind of a familiar puzzle, Magic Square (Bogomolny, 2006). Euler used *Latin* as the name of Latin Square because he used Latin characters as the symbols in a square (Wikipedia, 2006).

A *reduced* Latin Square or sometimes called *normalized* or *in standard form* if the elements in first column and first row are in natural order (Wikipedia, 2006). Figure 2.1 shows an example of reduced Latin Square and Figure 2.2 shows an example that is not a reduced Latin Square.

1	2	3	4
2	1	4	3
3	4	1	2
4	3	2	1

Figure 2.1 An example of reduced Latin Square



1	2	3	4
4	1	2	3
3	4	1	2
2	3	4	1

Figure 2.2 An example of not a reduced Latin Square

The Latin Square in Figure 2.1 is said to be reduced because the first row and first column also $\{1, 2, 3, 4\}$. The example in Figure 2.2 is not a reduced form because the first column is not in natural order. A not reduced Latin Square can be changed to reduced form by permuting or reordering (Wikipedia, 2006). This permuting or reordering concept is applied in this research to obtain different Latin Squares from an initial Latin Square.

When a game, especially a puzzle game becomes famous, the mathematician will try to calculate all the number of possible puzzle. For Latin Square, all the possible Latin Squares can be computed by using the following formula:

$$[(n!)(n-1)!] \times R$$

where n is the size of a Latin Square and R is the number of reduced Latin Square for certain size of Latin Square (Wikipedia, 2006). Table 2.1 shows the number of reduced Latin Square for different size.

The total number of possible Latin Square is needed to know so that the possible Latin Cube for a certain order can be calculated. For example, the possible Latin Cube for an order 4 Latin Square should not be more than 576 because a dimension is added. This is important when find a Latin Cube in certain order. The

total number can be estimated before the cube can be found. Table 2.2 shows the total number of Latin Square for $n = 1, \dots, 11$.

Table 2.1 The number of reduced Latin Square with different n

n	R
1	1
2	1
3	1
4	4
5	56
6	9408
7	16942080
8	535281401856
9	377597570964258816
10	7580721483160132811489280
11	5363937773277371298119673540771840

Table 2.2 Total number of Latin Square

n	Total number of Latin Square
1	1
2	2
3	12
4	576
5	161280
6	812851200
7	61479419904000
8	108776032459082956800
9	5524751496156892842531225600
10	9982437658213039871725064756920320000
11	776966836171770144107444346734230682311065600000



2.2 SUDOKU

This section will discuss about a brief history of SuDoku, some basic strategies to solve the SuDoku, total number of possible SuDoku grid and Dion Cube, which is where the idea of SuDoku Cube came from.

2.2.1 Brief History of SuDoku

SuDoku is a puzzle that is similar to Latin Square. This puzzle was designed by Howard Garns, a retired architect and freelance puzzle constructor (Maxwell, 2005). The main idea came from the Latin Square that created by Euler, but Howard Garns added an extra rule for it, which the entries in every $m \times m$ region must occur no more than once (Carter, 2006).

SuDoku puzzle was first published in 1979 at U.S. (Wikipedia, 2006). The SuDoku Puzzle was first published by the specialist puzzle publisher in New York, Dell Magazines, in its magazine Dell Pencil Puzzles and Word Games. They put the title as "Number Place" (Wikipedia, 2006). So, in U.S., SuDoku puzzle is also known as "Number Place" (Santos-Garc'ia & Palomino, 2006).

In April 1984, the SuDoku puzzle was first introduced in Japan by Nikoli, in the paper Monthly Nikolist and known as "Suuji wa dokushin ni kagiru" (Timmerman, 2006). SuDoku can be translated as "the numbers must be single" or "the numbers must occur only once" (Winkler, 2005). The puzzle was named by Kaji Maki, the



president of Nikoli. At a later date, the name was abbreviated to SuDoku where Su means number and Doku means single or odd (Wikipedia, 2006).

2.2.2 Strategies to Solve SuDoku

Usually, there are a few strategies to solve a SuDoku puzzle. The first strategy is scanning. Scanning consist two basic techniques, which are cross-hatching and counting. Cross-hatching, being the scanning of rows or columns, to identify which line in a particular region may contain a certain number (Lexico Publishing Group, 2006). Then, the process is repeated among the rows and the columns. Counting is used to identify a missing number in a row, column or an $m \times m$ region.

The second strategy is marking-up. This method is used when all the blank cells have more than one candidate. Then, a blank cell can be marked up with the possible candidate. In this situation, the last strategy can be used, which is analyzing. The what-if approach is using here. This means that a blank cell can be chosen and then make a guess to select a candidate number (Lexico Publishing Group, 2006). If the number that had chosen is the solution, then, the solving progress can be continued. Otherwise, these two strategies will be repeated.

SuDoku requires no math ability. It is a puzzle of logic and will not involve the mathematics operation such as adding and subtracting. So, players do not need to master in mathematics when they play the puzzle (Kingfisher Books, 2005).



2.2.3 Total Number of Possible SuDoku Grid

Similar to the Latin Square, many people are interested in finding the total number of possible SuDoku grid. Felgenhauer and Jarvis (2005) wrote a paper that enumerates the total number of possible SuDoku grids. In this paper, they calculated that SuDoku have 6670903752021072936960 possible grids.

This calculation is based on the standard 9×9 SuDoku. The formula to get this number showed as below:

$$9! \times 72^2 \times 2^7 \times 27,704,267,971 = 6670903752021072936960$$

$$\approx 6.671 \times 10^{21}$$

The last factor is a prime number (Wikipedia, 2006). This formula was derived by using logic theory. The total number of possible SuDoku grid is observed because the purpose is same with the purpose to know the total number of Latin Square.

2.2.4 Dion Cube

In May 2005, Dion Church, a Telegraph SuDoku enthusiast (Wikipedia, 2006) comes up with an “ultimate SuDoku”, which is a three dimension SuDoku. This 3D SuDoku is called the Dion cube. The entire rules that apply in 2D SuDoku also apply in 3D SuDoku. The Dion cube is first published in the Daily Telegraph in May 2005. (Taalman, 2006). Figure 2.3 shows an example of Dion Cube.



REFERENCES

- Bogomolny, A. 2006. *Latin Square*. Retrieved 30 May 2006 from <http://www.cut-the-knot.org/arithmetic/latin.shtml>.
- Carter, E. 2006. *The SuDoku Master – Advanced Strategies & Visual Techniques*. Lulu Press, Inc.
- Crownover, E. & Thibault, J. 2005. *Latin Square*. Retrieved 30 May 2006 from <http://web.usna.navy.mil/~wdj/teach/sm463/thibault-crownover-latin-squares.pdf>.
- Felgenhauer, B. & Jarvis, F. 2005. *Enumerating possible Sudoku grids*. Retrieved 31 May 2006 from <http://www.afjarvis.staff.shef.ac.uk/sudoku/sudoku.pdf>.
- Kingfisher Books. 2005. *SuDoku*. Kingfisher Publications Plc.
- Lexico Publishing Group, LLC. 2006. *Sudoku wikipedia encyclopedia reference*. Retrieved 30 May 2006 from <http://www.reference.com/browse/wiki/Sudoku>.
- Maxwell, K. 2005. *Macmillan English Dictionary Word of The Week Archive – Sudoku*. Retrieved 31 May 2006 from <http://www.macmillandictionary.com/New-Words/051107-Sudoku.htm>.
- Santos-García, G. & Palomino, M. 2006. *Solving Sudoku Puzzles with Rewriting Rules*. Electronic Notes in Theoretical Computer Science. Retrieved 30 May 2006 from www.elsevier.nl/locate/entcs.
- Taalman, L. 2006. *POW #11: 4x4x4 Sudokube*. Retrieved 31 May 2006 from <http://www.math.jmu.edu/~taal/POW/POW11.html>.



Timmerman, C. 2006. *THE EVERYTHING 30-MINUTE SUDOKU BOOK*. F+W Publications, Inc., United State.

Weisstein, E. 2006. *Latin Square*. Wolfram Mathworld. Retrieved 31 May 2006 from <http://mathworld.wolfram.com/LatinSquare.html>.

Wikipedia. 2006. *Latin Square*. Wikimedia Foundation, Inc. Retrieved 1 June 2006 from http://en.wikipedia.org/wiki/Latin_square.

Wikipedia. 2006. *Sudoku - Wikipedia, The Free Encyclopedia*. Retrieved 1 June 2006 from <http://en.wikipedia.org/wiki/Sudoku>.

Winkler, M. 2005. *Super Sudoku 360 Volume 1*. Brain Kandy Books, United State.

