International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 14 Issue 2 ǁ February 2026 ǁ PP. 32-37

New insights on primes with binary number matrices

Rodolfo Alvarez
Author contact: rodolfo.alv.zm@gmail.com

Abstract
The purpose of this article is going one step further, in the study of primes through binary numbers, in this case by refining the method of finding the amount of primes between 1 and n *within a binary matrix.*

Keywords: *Binary numbers, matrices, prime numbers.*

Date of Submission: 22-01-2026 Date of Acceptance: 04-02-2026

I. INTRODUCTION

Recent research on prime numbers, has shed light into some of their most relevant aspects (Chang, 2014; Mazurkin, 2014; Alvarez, 2024, 2025). One of them is certainly the amount of primes between 1 and *n*. In the case of our own research (Alvarez, 2024, 2025), we have been working on effective ways to find the amount of primes between 1 and n, by using a binary matrix. Next sections will develop these ideas in-depth.

II. THEORETICAL FRAMEWORK

2.1 Prime numbers

Prime numbers are those numbers that have two factors, i. e., one and the number itself (Kumar & Mozar, 2020). In the case of this research, they are found without the need of divisions or factorization, just the use of binary numbers within a matrix, as we can see in the following subsection.

2.1.1 Prime numbers within binary matrices

This article and its underlying research, are part of a sub-line of research on primes, based on binary numbers. Other scholars including us such as Chang (2014), Mazurkin (2014) and Alvarez (2024, 2025), have worked on primes using binary numbers. In this case we focus primarily on last author's work in the field, meaning our own research so as to yield a continuum in the search itself. In this particular article, we will work with binary numbers within binary matrices or tables.

III. DISCUSSION

To begin with, let us say we want to generate a 20x20 matrix to be filled with 1s and 0s, so as to get the amount of prime numbers between 1 and 20. First we set a semiperimeter of 1s as we can see in the matrix we are about to show. Second, we have a middle zone, basically what is inside the semiperimeter, and we need to fill it with the following sequential loops of 1s and 0s, in that very order:

1-0

1-1-0

1-1-1-0

1-1-1-1-0

…

, and so on. The initial representation of both steps is shown in the following matrix:

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	1	0																
		1	1	1	0														
			1	1	1	1	0												
				1	1	1	1	1	0										
					1	1	1	1	1	1	0								
						1	1	1	1	1	1	1	0						
							1	1	1	1	1	1	1	1	0				
								1	1	1	1	1	1	1	1	1	0		
									1	1	1	1	1	1	1	1	1	1	0
										1									
											1								
												1							
													1						
														1					
															1				
																1			
																	1		
																		1	
																			1

Matrix 1: Initial representation of binary matrix.

After that, we repeat the sequence of loops until the end of the matrix, in the same way:

1-0

1-1-0

1-1-1-0

1-1-1-1-0

…

And so on as shown in the following matrix:

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
			1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
				1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0
					1	1	1	1	1	1	0	1	1	1	1	1	0	1	0
						1	1	1	1	1	1	1	0	1	1	1	1	1	1
							1	1	1	1	1	1	1	1	0	1	1	1	1
								1	1	1	1	1	1	1	1	1	0	1	1
									1	1	1	1	1	1	1	1	1	1	0
										1									
											1								
												1							
													1						
														1					
															1				
																1			
																	1		
																		1	
																			1

Matrix 2: Modified binary matrix after the initial representation.

Additionally we fill the remaining squares with 1s, as a logical step:

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
			1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
				1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0
					1	1	1	1	1	1	0	1	1	1	1	1	0	1	0
						1	1	1	1	1	1	1	0	1	1	1	1	1	1
							1	1	1	1	1	1	1	1	0	1	1	1	1
								1	1	1	1	1	1	1	1	1	0	1	1
									1	1	1	1	1	1	1	1	1	1	0
										1	1	1	1	1	1	1	1	1	1

1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1
		1	1	1	1	1	1	1
			1	1	1	1	1	1
				1	1	1	1	1
					1	1	1	1
						1	1	1
							1	1
								1

Matrix 3. Full binary matrix representation.

After that, we multiply the values in each column thus yielding their corresponding outputs. This process takes an optimally minimized number of steps, since multiplying by 0 can happen very early in the column, the result being automatically 0 in that case.

1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
			1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
				1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0
					1	1	1	1	1	1	0	1	1	1	1	1	0	1	0
						1	1	1	1	1	1	1	0	1	1	1	1	1	1
							1	1	1	1	1	1	1	1	0	1	1	1	1
								1	1	1	1	1	1	1	1	1	0	1	1
									1	1	1	1	1	1	1	1	1	1	0
										1	1	1	1	1	1	1	1	1	1
											1	1	1	1	1	1	1	1	1
												1	1	1	1	1	1	1	1
													1	1	1	1	1	1	1
														1	1	1	1	1	1
															1	1	1	1	1
																1	1	1	1
																	1	1	1
																		1	1

																			1
-	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0

Matrix 4. Full binary matrix representation with prime number information.

As we can see, as first 1 cannot multiply anything it is automatically discarded, and the rest speaks for itself. If we focus on last row and list it using orderly natural numbers we will obtain the following extended matrix:

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
		1	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1
			1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
				1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	0
					1	1	1	1	1	1	0	1	1	1	1	1	0	1	0
						1	1	1	1	1	1	1	0	1	1	1	1	1	1
							1	1	1	1	1	1	1	1	0	1	1	1	1
								1	1	1	1	1	1	1	1	1	0	1	1
									1	1	1	1	1	1	1	1	1	1	0
										1	1	1	1	1	1	1	1	1	1
											1	1	1	1	1	1	1	1	1
												1	1	1	1	1	1	1	1
													1	1	1	1	1	1	1
														1	1	1	1	1	1
															1	1	1	1	1
																1	1	1	1
																	1	1	1
																		1	1
																			1
-	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	1	0
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20

Matrix 4. Full binary matrix representation with prime number information, connected to each natural number.

, in which each 1 corresponds to a prime number between 1 and 20, and there are 8 of them in total within that range. By extending this matrix to bigger numbers, we can use these findings as a computational or mathematical tool to make faster calculations.

IV. CONCLUSION

In this publication, we developed new insights on the amount of primes between 1 and n, by using binary number matrices. The methods used in previous research have been simplified and clarified, and this article will hopefully be useful in future research.

REFERENCES

[1]. Alvarez, R. (2024). A simple method for finding prime numbers: An infinite model based on binary numbers. *International Journal of Scientific and Engineering Research.* Vol. 15, No. 12.

[2]. Alvarez, R. (2025). Prime numbers, binary matrices and generation rules. *International Journal of Research in Engineering and Science.* Vol. 13, No. 10.

[3]. Chang, J. (2014). Every complete binary tree is prime. *International Journal of Computer Applications.* Vol. 106, No. 14.

[4]. Kumar, A. & Mozar, S. (2020). *Proceedings of the 3rd International Conference on Communications and Cyber Physical Engineering*, p. 170. New York: Springer Nature.

[5]. Mazurkin, P. (2014). Series primes in binary. *American Journal of Applied Mathematics and Statistics.* Vol. 2, No. 2, pp. 60-65. Science and Education Publishing.