On the Generators of the Group of Units Modulo a Prime and Its Analytic and Probabilistic Views

Ricky B. Villeta[1]*, Elmer C. Castillano[2], and Roberto N. Padua[3]

[1,2,3] University of Science and Technology of Southern Philippines, Cagayan de Oro City, Philippines

[1] https://orcid.org/0000-0001-8386-6798

[2] https://orcid.org/0000-0002-1230-8817

[3] https://orcid.org/0000-0002-2054-0835

*Email Correspondence: rbvilleta35@usjr.edu.ph

Abstract

This paper further investigates the cyclic group $(Z_p)^*$ with respect to the primitive roots or generators $g \in (Z_p)^*$. The simulation algorithm that determines the generators and the number of generators, g *of* $(Z_p)^*$ *for a prime* p *is illustrated using Python programming. The probability of getting a generator* g *of* $(Z_p)^*$, *denoted by* $\frac{\phi(\phi(p))}{\phi(p)}$, *is generated for prime p between 0 to 3000. The scatterplot is also shown that depicts the data points on the probability* $\frac{\phi(\phi(p))}{\phi(p)}$ *of the group of units* $(Z_p)^*$ *with respect to the order* p - 1 *of* $(Z_p)^*$ *for prime* p *between 0 to 3000. The scatterplot results reveal that the probability of getting a generator of the group of units* $(Z_p)^*$ *is fluctuating within the probability range of 0.20 to 0.50, for prime p modulus from 3 to 3000. These findings suggest that the proportion of the number of generators of the group of units modulo a prime of order* p - 1, *though fluctuating, is bounded from 20% to 50% for prime* p *modulus from 3 to 3000.*

Keywords: Group of units modulo a prime, $(Z_p)^*$, *primitive roots or generators of* $(Z_p)^*$, *simulation algorithm, probability of getting a generator* g *of* $(Z_p)^*$ *.*

1.0 Introduction

Let Z_n be the set of integers *{0, 1, 2, ..., n - 1}* under addition modulo n. Then the set of all elements a of Z_n relatively prime to n, that is, $\gcd(a, n) = 1$, under multiplication modulo n forms a group denoted by $(Z_n)^*$. The order of this group, $\left|(Z_n)^*\right|$, is equal to $\phi(n)$ where:

$$\phi(n) = n \prod_{p/n} \left(1 - \frac{1}{p}\right).$$

The function ϕ is called the Euler Totient function (Vinogradov, 2003).

The group $(Z_n)^*$ is cyclic if and only if n is equal to 1, 2, 4, p^k or $2p^k$ (Gauss, 1966). When n = p is prime, it follows that $(Z_n)^*$ is a cyclic group of order $\phi(p) = p-1$.

A number g is a generator of a cyclic group under multiplication modulo *n*, if for each *b* in this group, there exists a *k*, such that $g^k \equiv b(\text{modulo } n)$, $\gcd(b,n)=1$. Such a generator is called a primitive root modulo *n*. The integer *k* is called the index of *b* to the base g modulo *n* (sometimes referred to as the discrete logarithm of *b* to the base g modulo *n*). When *n* = *p* is a prime, the number of primitive roots modulo *n* is $\phi(\phi(p)) = \phi(p-1)$, since a cyclic group of *p* - *1* elements has $\phi(p-1)$ generators (Vinogradov, 2003). Knuth (1998) showed that:

$$\frac{n}{\phi(n-1)} = O(\log \log n)$$

so that for large *n*, the generators are very common among {2, 3, . . . , *n* - *1*}.

This study endeavors to investigate further the cyclic group $(Z_p)^*$, and the elements of $(Z_p)^*$, specifically the generators $g \in (Z_p)^*$. The simulation algorithm that determines the generators and the number of generators, g of $(Z_p)^*$ for a prime *p* is illustrated using the Python programming. The distribution of the resulting number of generators for each prime *p* as modulus of the cyclic group $(Z_p)^*$ is presented using a scatterplot diagram. The probability of getting a generator g of $(Z_p)^*$, denoted by $\frac{\phi(\phi(p))}{\phi(p)}$ is also generated for prime *p* between 1 and 3000.

2.0 Prime Generators of $(Z_p)^*$

The group $(Z_p)^*$ under modulo *p* is cyclic with $\phi(p) = p-1$ elements. The number of generators of this cyclic group, therefore is, at most $\phi(\phi(p)) = \phi(p-1)$ (Vinogradov, 2003). We enumerated facts about the generators of $(Z_p)^*$ and had proven some of them. Wilson's Theorem (Burton, 2007, p. 94) in number theory is an important tool in deriving a result for the product of generators g_i of $(Z_p)^*$ for a prime *p*. It says:

Theorem 2.1 (Wilson) Let *p* be a prime number. Then $(p-1)! \equiv -1 \text{ modulo } p$.

While Wilson's result can be used as a primality test, however, it is computationally intractable. It remains an important theoretical result. Next, if *p* is a prime, then $(Z_p)^*$ has $\phi(p) = p-1$ elements. Since $(Z_p)^*$ is cyclic, it has $\phi(p-1)$ generators.

Examples 2.2

(1) If p = 11, $(Z_{11})^*$ has $\phi(11) = 10$ elements and it has $\phi(10) = \phi(\phi(11))$ generators, that is, $\phi(10) = 4$. The generators are {2,6,7,8}. Note that $2 \cdot 6 \cdot 7 \cdot 8 \equiv 1(\text{mod } 11)$ since $2 \cdot 6 \equiv 1(\text{mod } 11)$ and $7 \cdot 8 = 56 \equiv 1(\text{mod } 11)$.

(2) If p = 17, $(Z_{17})^*$ has $\phi(17) = 16$ elements, and it has $\phi(\phi(17)) = \phi(16) = 8$ generators, namely, {3, 5, 6, 7, 10, 11, 12, 14}. We can re-group generators as follows {(3,6), (5,7), (10,12), (11,14)}, so that $\prod_{i=1}^{8} g_i \equiv 1(\text{mod } 17)$.

The following result shows that the product of generators g_i of the group of units modulo a prime p is congruent to 1 modulo p. Fermat's Theorem (Burton, 2007, p. 88) is used to prove this result.

Theorem 2.3 (Fermat's Theorem) Let p be a prime and suppose that p does not divide a. Then, $a^{p-1} \equiv 1 \pmod{p}$.

Theorem 2.4 Let p be a prime. Then $(Z_p)^*$ has $\phi(p-1)$ generators and

$$\prod_{i=1}^{\phi(p-1)} g_i \equiv 1 \pmod{p}.$$

Proof: The first part follows from the fact that $(Z_p)^*$ has $\phi(p) = p-1$ elements. Since $(Z_p)^*$ is cyclic then, it has $\phi\,(p - 1)$ generators. Next, take a generator g_k. By Fermat's Theorem (Theorem 2.3),

$$g_k^{p-1} \equiv 1 \pmod{p} \text{ for } k = 1, 2, \cdots, \phi(p-1).$$

For each j, $g_j = g_k^{d_j}$ since g_k is a generator. Now,

$$\prod_{j=1}^{\phi(p-1)} g_j = \prod_{j=1}^{\phi(p-1)} g_k^{d_j} = g_k^{d_1} g_k^{d_2} \dots g_k^{d_{\phi(p-1)}}$$
$$= g_k^{d_1+d_2+\cdots+d_{\phi(p-1)}} = g_k^{\sum_{j=1}^{\phi(p-1)} d_j}.$$

We can pair each term by their inverses and this gives:

$$\prod_{i=1}^{\phi(p-1)} g_i = g_k^{\frac{\phi(p-1)}{2}(\phi(p))} \equiv 1 \pmod{p}. \ \blacksquare$$

Consider, next, the prime factors of $\phi\,(p)$ where p is a prime. Suppose that $\phi(p) = 2p_1 p_2 \cdots p_k$. Let Q be the set of all primes less than or equal to p, $Q = \{q_1, q_2, \cdots, q_m\}$. Then, it is clear that $\{p_1, p_2, \cdots, p_k\} \subseteq Q \subseteq (Z_p)^*$.

Lemma 2.5 Let Q be the set of all primes less than or equal to p and let P be the set of all prime factors of $\phi(p)$. Then $P \subseteq Q \subseteq (Z_p)^*$.

Proof: Let $p_j \in P$, then $p_j / \phi(p)$ and so $p_j < p$. Moreover, $\gcd(p_j, p) = 1$, hence, $p_j \in Q \subseteq (Z_p)^*$. It follows that $P \subseteq Q$. $\blacksquare$

3.0 Analytic and Probabilistic Procedure in Finding Generators of the Cyclic Group, $(Z_p)^*$

An element of the group of units modulo a prime p, $g \in (Z_p)^*$ is a generator if $(Z_p)^* = \{g^k : k \in Z\}$. The computation of generators of the cyclic group, $(Z_p)^*$ is indispensable in pseudorandom number generators, error detecting codes, and in many cryptosystems such as the following: Diffie-Hellman key exchange protocol; ElGamal and Massey-Omura public key ciphers; DSA; ElGamal and Nyberg-Rueppel digital signature (Adamski & Nowakowski, 2015).

The following result, Theorem 3.1, Adamski & Nowakowski (2015), in algebraic number theory is useful in the simulation algorithm which can be used to obtain the generators of the cyclic group, $(Z_p)^*$ modulo a prime p.

Theorem 3.1 Let $(Z_p)^*$ be the cyclic group of the group of units modulo a prime p of order $\phi(p) = p-1$. Let $2p_1 \cdot p_2 \cdots p_k$ be the prime factorization of $\phi(p)$. Then, $g \in (Z_p)^*$ is a generator of $(Z_p)^*$ if and only if for all $i = 1, 2, \ldots, k$, $g^{\frac{\phi(p)}{p_i}}$ is not congruent to 1 modulo p.

Simulation Algorithm for Finding Generators of the Group of Units Modulo a Prime

This section determines the simulation algorithm that constructs a large prime p for the modulus of $(Z_p)^*$, and finds the generator and the number of generators of $(Z_p)^*$. Python programming was used in the implementation of this algorithm.

Constructing the Large Prime *p* for the Modulus of $(Z_p)^*$

In constructing the large prime p for the modulus of $(Z_p)^*$, the Miller-Rabin Test (Rabin, 1980) for the test of primality can be used.

The Miller-Rabin Test of Primality

Suppose n is prime with $n > 2$, hence $n - 1$ is even, which can be written as $2^t e$, where t and e are positive integers (e is odd). For each integer x, $1 < x < n$, then either $x^e \equiv \pm 1 (\mathrm{mod}\, n)$ or $x^{2^r e} \equiv -1 (\mathrm{mod}\, n)$ for any r with $1 \leq r \leq t-1$.

The Miller-Rabin primality test is the contrapositive of the preceding statement, that is, in the event that we can find an x^e is not congruent to 1or -1 (mod n) or $x^{2^r e}$ is not congruent to -1 mod (n), for all $1 \leq r \leq t-1$, then n is not prime.

Finding the Generators $g \in (Z_p)^*$ for a Prime *p*

The following outlines the simulation algorithm for finding the generators $g \in (Z_p)^*$ for a large prime p as the modulus of $(Z_p)^*$:

1. Determine the number n if prime using the Miller-Rabin primality test. If n is prime, then denote it by p;
2. Get the prime factors of p-1, that is, $\phi(p) = p - 1 = 2 p_1 \cdot p_2 \cdots p_k$;
3. Initialize the list of generator;
4. Iterate j from 1 to $\phi(p) = p - 1$, the order or size of $(Z_p)^*$;
5. In every iteration j, initialize flag to a generator;
6. Iterate i for all the prime factors of $\phi(p) = p - 1$;
7. If $j^{\left(\frac{p-1}{i}\right)} \equiv 1 (\mathrm{mod}\, p)$, then make a flag that j is not a generator;
8. Outside the iteration of the prime factors, provide a condition for checking the flag;
9. If flag is true, then j is a generator and append to the list of generators of $(Z_p)^*$;
10. Count the number of generators of $(Z_p)^*$ in the list; and
11. Iterate steps 1 to 10 to generate all the generators of $(Z_p)^*$, for prime p between 1 and 3000.

4.0 Simulation Results for the Generators and Number of Generators of the Group of Units Modulo a Prime for Prime Modulus Between 0 to 3000

Figures 1, 2, 3, 4 and 5 depict the scatterplot for the data points on the number of generators of the group of units $(Z_p)^*$ versus the corresponding prime number modulus from 0 to 3000.

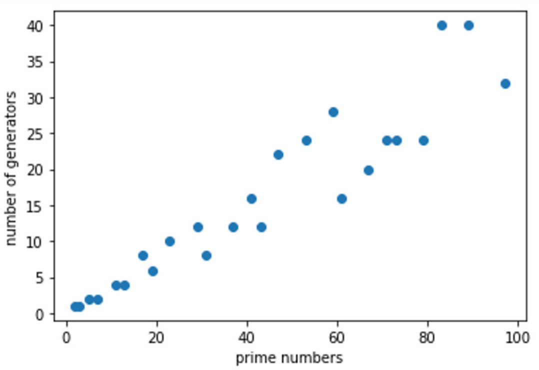

Figure 1. *Scatterplot for the number of generators of* $\left(Z_p\right)^*$ *versus the corresponding prime number modulus between 0 and 100*

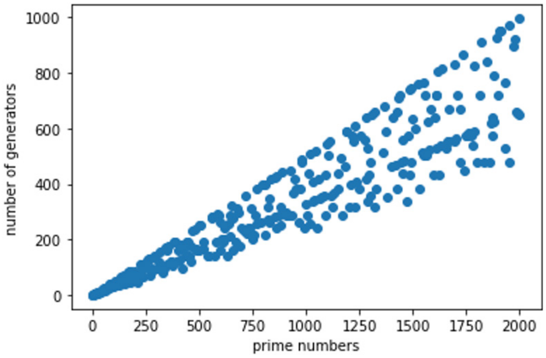

Figure 4. *Scatterplot for the number of generators of* $\left(Z_p\right)^*$ *versus the corresponding prime number modulus between 0 and 2000*

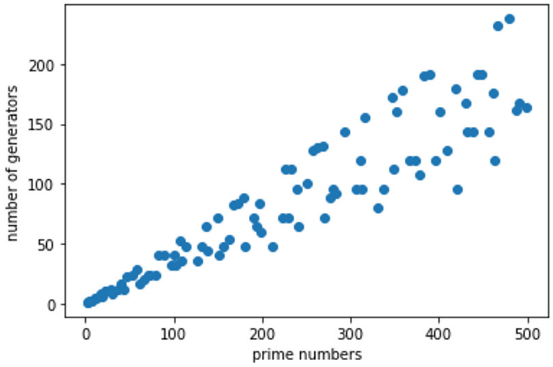

Figure 2. *Scatterplot for the number of generators of* $\left(Z_p\right)^*$ *versus the corresponding prime number modulus between 0 and 500*

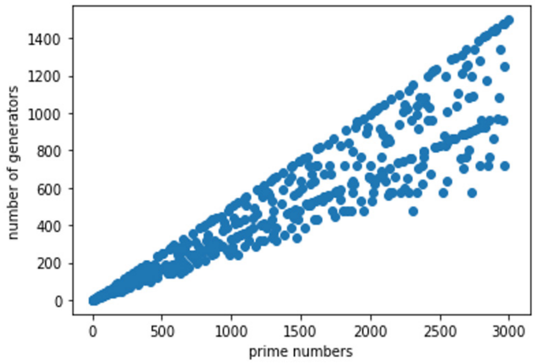

Figure 5. *Scatterplot for the number of generators of* $\left(Z_p\right)^*$ *versus the corresponding prime number modulus between 0 and 3000*

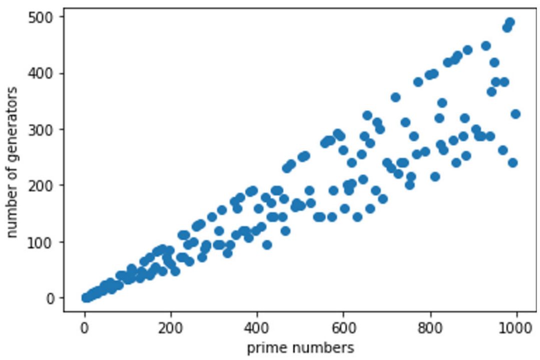

Figure 3. *Scatterplot for the number of generators of* $\left(Z_p\right)^*$ *versus the corresponding prime number modulus between 0 and 1000*

5.0 The Probability, $\frac{\phi(\phi(p))}{\phi(p)}$ Behavior of Finding a Generator of the Group of Units Modulo a Prime p for each Prime Modulus Between 0 to 3000

Figures 6, 7, 8, 9 and 10 depict the scatterplot for the data points on the probability $\frac{\phi(\phi(p))}{\phi(p)}$ of the group of units $\left(Z_p\right)^*$ versus the corresponding order *p-1* of $\left(Z_p\right)^*$ for prime p between 2 to 3000. The scatterplot results reveal that the probability of getting a generator of the group of units $\left(Z_p\right)^*$

is fluctuating within the probability range of 0.20 to 0.50, for prime p modulus from 3 to 3000. These findings suggest that the proportion of the number of generators of the group of units modulo a prime of order p - 1, though fluctuating, is bounded from 20% to 50% for prime p modulus from 3 to 3000.

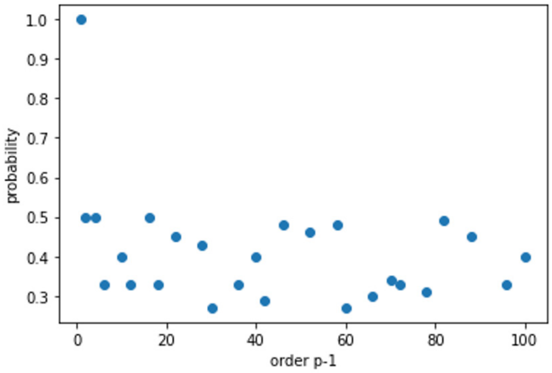

Figure 6. *Scatterplot for the probability* $\frac{\phi(\phi(p))}{\phi(p)}$ *of the group of units* $(Z_p)^*$ *versus the corresponding order p-1 of* $(Z_p)^*$ *for prime p between 0 and 100*

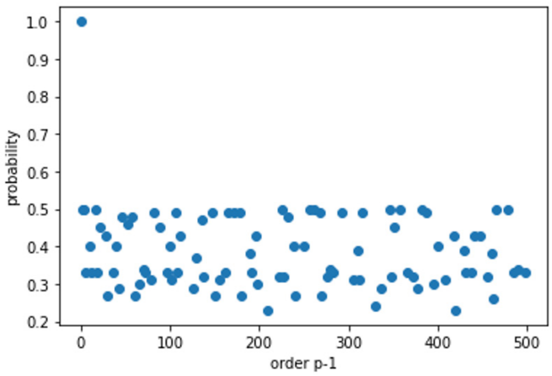

Figure 7. *Scatterplot for the probability* $\frac{\phi(\phi(p))}{\phi(p)}$ *of the group of units* $(Z_p)^*$ *versus the corresponding order p-1 of* $(Z_p)^*$ *for prime p between 0 and 500*

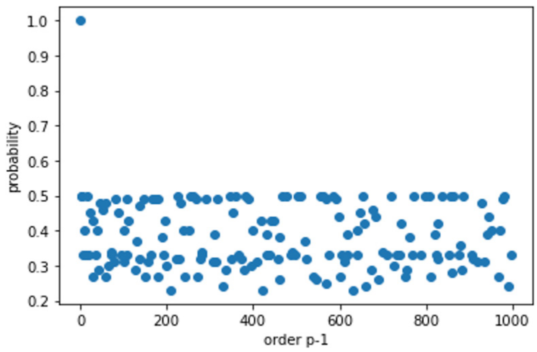

Figure 8. *Scatterplot for the probability* $\frac{\phi(\phi(p))}{\phi(p)}$ *of the group of units* $(Z_p)^*$ *versus the corresponding order p-1 of* $(Z_p)^*$ *for prime p between 0 and 1000*

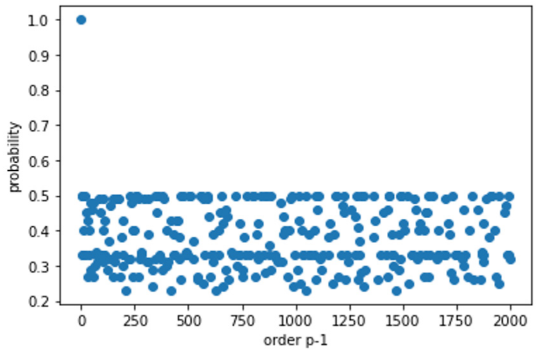

Figure 9. *Scatterplot for the probability* $\frac{\phi(\phi(p))}{\phi(p)}$ *of the group of units* $(Z_p)^*$ *versus the corresponding order p-1 of* $(Z_p)^*$ *for prime p between 0 and 2000*

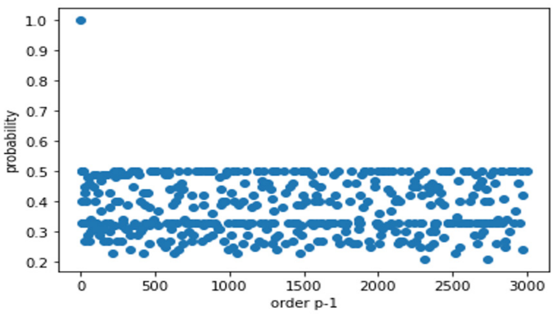

Figure 10. *Scatterplot for the probability* $\frac{\phi(\phi(p))}{\phi(p)}$ *of the group of units* $(Z_p)^*$ *versus the corresponding order p-1 of* $(Z_p)^*$ *for prime p between 0 and 3000*

6.0 Conclusion

This study investigated further the cyclic group $(Z_p)^*$ with respect to the primitive roots or generators $g \in (Z_p)^*$. The simulation algorithm that determines the generators and the number of generators, g of the cyclic group $(Z_p)^*$, for prime p is illustrated using the Python programming. The probability of getting a generator g of $(Z_p)^*$ denoted by $\frac{\phi(\phi(p))}{\phi(p)}$ is generated for prime p between 0 to 3000. The scatterplot results for the data points on the probability $\frac{\phi(\phi(p))}{\phi(p)}$ of the group of units $(Z_p)^*$ with respect to the order p - 1 of $(Z_p)^*$ reveal that the probability of getting a generator of the group of units $(Z_p)^*$ is fluctuating within the probability range of 0.20 to 0.50 for prime p modulus from 3 to 3000. These findings suggest that the proportion of the number of generators of the group of units modulo a prime of order p - 1, though fluctuating, is bounded from 20% to 50% for prime p modulus from 3 to 3000.

References

Adamski, T., & Nowakowski, W. (2015). The average time complexity of probabilistic algorithms for finding generators in finite cyclic groups. *Bulletin of the Polish Academy of Sciences,Technical Sciences, 63*(4), 989-996. https://doi.org/10.1515/bpasts-2015-0112.

Burton, D. M., (2007). *Elementary number theory* (6thed.). McGraw-Hill.

Gauss, C. F. (1966). *Disquisitiones arithmeticae* (English ed). Springer-Verlag. https://doi.org/10.1007/978-1-4939-7560-0.

Knuth, D. E. (1998). *The art of computer programming: Vol. 2. Seminumerical algorithms* (3rd ed.). Addison-Wesley.

Rabin, M. O. (1980). Probabilistic algorithm for testing primality. *Journal of Number Theory, 12*(1), 128-138. https://doi.org/10.1016/0022-314X(80)90084-0.

Vinogradov, I. M. (2003). *Elements of number theory*. Dover Publications Inc. https://books.google.com.ph/books?id=xllfdGPM9t4C&printsec=frontcover#v=onepage&q&f=false