[Egyptian Informatics Journal 22 (2021) 101–104](https://doi.org/10.1016/j.eij.2020.05.002)

Contents lists available at [ScienceDirect](http://www.sciencedirect.com/science/journal/11108665)

Egyptian Informatics Journal

journal homepage: [www.sciencedirect.com](http://www.sciencedirect.com/)

Full length article

[](http://crossmark.crossref.org/dialog/?doi=10.1016/j.eij.2020.05.002&domain=pdf)A new explicit algorithmic method for generating the prime numbers in order

Abd Elhakeem Abd Elnaby [a](#_bookmark0), A.H. El-Baz [b](#_bookmark1)

a *Department of Mathematics, Faculty of Science, Damietta University, New Damietta, Egypt*

b *Department of Computer Science, Faculty of Computers and Information, Damietta University, New Damietta, Egypt*

# a r t i c l e i n f o

*Article history:*

Received 9 October 2019

Revised 2 March 2020

Accepted 11 May 2020

Available online 11 June 2020

*Keywords:* Prime numbers Cryptography Algorithms

# a b s t r a c t

This paper presents a new method for generating all prime numbers up to a particular number *m* ∈ *N*; *m* P 9; by using the set theory. The proposed method is explicit and works oriented in finding the prime numbers in order. Also, we give an efficiently computable explicit formula which exactly deter- mines the number of primes up to a particular number *m* ∈ *N*; *m* P 9. For the best of our knowledge, this is the first exact formula given in literature. For the sake of comparison, a unified framework is done not

only for obtaining explicit formulas for the well-known sieves of Eratosthenes and Sundaram but also for

obtaining exact closed form expression for the number of generated primes using these two sieve meth- ods up to a particular number *m* ∈ *N*; *m* P 9. In addition, the execution times are calculated for the three methods and indicate that our proposed method gives a superior performance in generating the primes.

© 2020 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intel- ligence, Cairo University. This is an open access article under the CC BY-NC-ND license ([http://creative-](http://creativecommons.org/licenses/by-nc-nd/4.0/)

[commons.org/licenses/by-nc-nd/4.0/](http://creativecommons.org/licenses/by-nc-nd/4.0/)).

1. Introduction

Number theory has many increasingly important applications in computer science and cryptography [[1]](#_bookmark16). One of the central topics of number theory is the prime numbers. The prime numbers are natural numbers greater than 1, which have no positive divisors other than 1 and itself but the natural numbers greater than 1 that are not prime are called composite numbers.

Generating prime numbers by using sieving methods has played a vital role in applied number theory. There are two well- known approaches for draining the composite numbers and leav- ing the prime numbers by sieving namely, the sieves of Eratos- thenes and Sundaram [[2]](#_bookmark16) that have been used for obtaining all prime numbers up to a particular number.

In this study, we propose a direct method which depends on the concepts of the sets to generate all the prime numbers in order. Moreover, it is well-known that the prime numbers theorem [[2]](#_bookmark16) approximately gives number of primes and which stated that if

p(*x*) denotes the number of primes up to a particular number *x*

pose a unified framework for obtaining explicit formulas for both the sieves of Eratosthenes and Sundaram. Also, exact explicit closed form formulas for the generated primes using these two sieves methods are obtained.

The reset of the paper is organized as follows: In [Section 2](#_bookmark2), the proposed method for generating the prime numbers is given along with its algorithm. In [Section 3](#_bookmark9), a unified framework is proposed to obtain the sieves of Eratosthenes and Sundaram using our pro- posed method. Conclusion is summarized in [Section 4](#_bookmark17).

1. The proposed method

We use set theory [[3]](#_bookmark16) to generate the prime numbers up to a particular number *m N*; *m* P 9; the proposed method is given by the following theorem:

∈

Theorem:

Let *P*(*m*) be the set of primes up to a particular number

*m* ∈ *N*; *m* P 9. If *A* ={(2*i* +1)(2*i* +1+2*n* ): *n* = 0; 1; 2; 3;

then p(*x*) ~ *x*/ln(*x*), however, using our proposed method, we give

j 2 k

*i i* *i*

*k*

an explicit formula which exactly determines the number of primes up to a particular number *m* ∈ *N*; *m* P 9. Besides, we pro-

...; *m*—(2 *i*+1)

2(2 *i*+1)

*i* = 1; 2; 3; ...*k*, *A* = S *Ai*; and *B* ={2*j* +1 : *j* = 1;

2; 3; ...; *m*—1 }; then *A*

*i*=1

2

*k*+1

*k*+1

2(2 *k*+3)

=u whenever max(*n*

) = j*m*—(2 *k*+3)2 k< 0;

*P*(*m*) = {2} ∪(*B* — *A*) and

*P*(*m*) =|*B*|— |*A*|+1; where |·| denotes the

*E-mail address:* [ali\_elbaz@yahoo.com](mailto:ali_elbaz@yahoo.com) (A.H. El-Baz)

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

cardinality.

<https://doi.org/10.1016/j.eij.2020.05.002>

1110-8665/© 2020 THE AUTHORS. Published by Elsevier BV on behalf of Faculty of Computers and Artificial Intelligence, Cairo University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

102 *A. Elhakeem Abd Elnaby, A.H. El-Baz / Egyptian Informatics Journal 22 (2021) 101–104*

Proof:

Firstly, it must be noted that *ni* determines number of generated elements in the set *Ai*, and *i* determines number of generated sets from the set *Ai*.

Since max(*n* ) = j*m*—(2 *k*+3 2 k < 0, then min(*n* ) < 0 which

*k*+1

)

*k*+1

Since *B* = (*B* — *A*) ∪ *A* and (*B* — *A*) ∩ *A* = u, then

|*B*| = |*B* — *A*| + |*A*| ⇒ |*B* — *A*| = |*B*| — |*A*|. (1.12)

Substitution from [(1.12)](#_bookmark3) in [(1.11)](#_bookmark13) gives

*P*(*m*)

= 1 + |*B*| — |*A*|. (1.13)

2 2 *k*+3)

(

shows no generated elements in the set *Ak*+1 because *nk*+1does not satisfy the condition on *ni* as stated in the set *Ai*. This gives

This proves the third requirement and hence the proof is

*Ak*+1 = u. This proves the first requirement.

Since max(*ni*)decreases with increasing *i* and the set

*Ak*+1 = u, then the set *Ak*contains at least one element and this satisfies only when max(*nk*) = min(*nk*) = 0. Consequently,

j 2 k j, k

*m*—(2 *k*+1) = 0 ⇒ *k* = ﬃ*m*ﬃﬃ—1 because *k* is positive integer.

2(2 *k*+1)

2

completed.

* Pseudocode for the proposed method

The following algorithm gives our proposed method for gener-

ating prime numbers.

Assume *Ai* and *A* are as stated and properties and definition of the floor function, we find that minimum and maximum elements

Algorithm: The proposed method for generating primes

in *A* =

*k*

*Ai*,

S

*i*=1

are 9 and 2 j,ﬃ*m*ﬃﬃ—1k

+ 1 2 respectively because mini-

1: function Prime (*m*) d *m* is the limit up to which

primes are generated

mum elements in *A*1 is 9 and maximum elements in *Ak* is

2

2 j,ﬃ*m*ﬃﬃ—1k + 1 2 . Moreover,

2

2: *i* ← 1

3: while *i* 6 *m*—1 do

2

## 2 ,,ﬃ*m*ﬃﬃﬃ — 1, + 1

2

2

6 *m*, when ,ﬃ*m*ﬃﬃﬃ is a positive real number.

4: *B*(*i*) 2*i* + 1

5: *i i* + 1

←

←

6: end while

Then the set of odd composite numbers *A* can be defined as follows:

## *A* = {9 6 *c* 6 *m* : *c* is an odd composite number} (1.1)

Assume *B* as stated and use properties of the floor function, we find that minimum and maximum elements in B are 3 and

7: *k* 1

8: *i* 1

←

←

9: while *k* > 0 do

← j k

10: *k m*—(2*i*+1)2

4*i*+2

11: *i i* + 1

←

12: end while

2  *m*—1

2

+ 1 respectively. Moreover, *m*—1 =

*m*—1 , *m* is odd

2 — 1, *m* is even

*m*

2

13: *k* ← *i*

14: *i* ← 1

Suppose *C* C *B* and *P* C *B* which are defined as follows:

2

## *C* = {9 6 *c* 6 *m* : *c* is an odd composite number}. (1.2)

And

## *P* = {3 6 *p* 6 *m* : *p* is a prime number}. (1.3)

Then B can be written as follows:

*B* = *C* ∪ *P*, (1.4)

Also, *P*(*m*) as stated can be defined as follows:

## *P*(*m*) = {2 6 *p* 6 *m* : *p* is a prime number}. (1.5)

It is clear from [(1.1) and (1.2)](#_bookmark4) that A = C Hence, [(1.4)](#_bookmark7) becomes

*B* = *A* ∪ *P*. (1.6)

Moreover, it is clear from [(1.2) and (1.3)](#_bookmark5) that *C P* u.

∩ =

Therefor

*A* ∩ *P* = u. (1.7)

From [(1.6) and (1.7)](#_bookmark8), we have

*P* = *B* — *A* (1.8)

Also, it is clear from [(1.3) and (1.5)](#_bookmark6) that

## *P*(*m*) = {2} ∪ *P*. (1.9)

Substitution from [(1.8)](#_bookmark10) in [(1.9)](#_bookmark11), we obtain

## *P*(*m*) = {2} ∪ (*B* — *A*). (1.10)

This proves the second requirement. Since (*B* — *A*) ∩ {2} = u, then [(1.10)](#_bookmark12) gives

## *P*(*m*) = |{2}| + |*B* — *A*| = 1 + |*B* — *A*|. (1.11)

15: *n* ← 0

16: while *i* 6 *k* do

← j k

17: *d m*—(2*i*+1)2

4*i*+2

18: while *n* 6 *d* do

19: *x n* 1 2*i* 1 2*i* 2*n* 1

( + )← ( + )( + + )

20: end while

21: *A x* d save the generated elements *x* in vector *A*

←

22: *x*

←[]

23: end while

24: *P = B-A* d *P* is the primes which represents the set difference of *B* and *A*

25: end function

1. A unified framework

For the sake of comparison with our proposed technique for generating primes, we proposed a unified framework to obtain the well-known sieving methods, namely the sieves of Eratos- thenes and Sundaram.

* 1. *Proposed technique for obtaining the sieve of Eratosthenes*

In order to obtain the sieve of Eratosthenes using our proposed method for generating all primes up to a particular number,

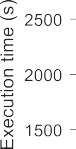
*m* ∈ *N*, *m* P 9, we perform the following steps:

Step 1: Let *BE* be the set of numbers less than or equal a partic- ular number *m,* i.e.

*BE* = {*j* + 1 : *j* = 1, 2, 3, .. . , *m* — 1}.

* + 1. *Elhakeem Abd Elnaby, A.H. El-Baz / Egyptian Informatics Journal 22 (2021) 101–104* 103

Step 2: Let *CE* be the set of even numbers greater than two and less than or equal a particular number *m,* i.e.



*CE* = n2*l* : *l* = 2, 3, ..., j*m*k o.

2

Step 3: Generate the set *AE* where

*i*

*AE* = (2 *i* + 1)(2 *ni* + 1) : *ni* = 1, 2, .. . , ,*m* — (2*i* + 1), ,

*i*

## *i* = 1, 2, 3, ... , *k*.

2(2*i* + 1)

where *AE* = j*m*—(2*i*+1)k,|.∫ is the Floor function and

| . | is the

*i*

Cardinality.

= + j k =

2(2*i*+1)

*AE.*

*i*

If *i k* 1, we obtain *m*—(2*i*+1) 0 then stop generating the set

2(2*i*+1)

*k*

Step 4: Let *A* = S *A*

*E E*

*i*

*i*=1

Step 5: Let *P*(*m*) be the set of primes up to a particular number

*E*

*m* ∈ *N*, *m* P 9, then

*P*(*m*) = *BE* — *AE* ∪ *CE* .

*E*



Fig. 1. The execution time versus a particular number *m* ∈ *N*, *m* P 9 for the proposed, Eratosthenes and Sundaram methods.

Also, the cardinality of *P*(*m*) is given by

*E*

Step 3: Generate the set *AS* where

*i*

*P*(*m*) *BE*

*AE*

*CE*

## *E* =

— ∪

*AS* = {*i* + *ni*(2 *i* + 1) : *ni* = 1, 2, 3, ...

*m*—(2 *i*+1) *i* 1 2 *k* And

j k}, = , , .. . , .

*i*

2 (2 *i*+1)

* 1. *Proposed technique for obtaining the sieve of Sundaram*

*AS* = j*m*—(2*i*+1)k,

|.∫ is the Floor function and

| . | is the

*i*

= + j k =

2(2*i*+1)

In order to obtain the sieve of Sundaram using our proposed method for generating all primes up to a particular number,

*m* ∈ *N*, *m* P 9, we perform the following steps:

Step 1: Let *BS* be the set of numbers less than or equal *m*.

Cardinality.

If *i k* 1, we obtain *m*—(2*i*+1) 0 then stop generating the

2(2*i*+1)

set *AS.*

*i*

*BS* = {*j* : *j* = 1, 2, 3, .. . , *m*}.

*k*

*S S*

Step 4: Let *A* = S *A*

*i*

*i*=1

Step 5: Let *BS* — *AS* ∪ *CS*

= {*r* : *r* ∈ *N*}

Step 2: Generate the set *CS* where

*CS* = nj*m*k + *q* : *q* = 0, 1, 2, .. . , *m* — j*m*ko

2

2

Step 6: Let *P*(*m*) be the set of primes up to a particular number

*m* ∈ *N*, *m* P 9, then

*s*

Table 1

The execution time versus a particular number *m* ∈ *N*, *m* P 9 for the proposed, Eratosthenes and Sundaram methods.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| m | Number of generated sets (k) |  |  |  | Execution time (in second) |  | | |
|  | Proposed method | Eratosthenes | Sundaram |  | Proposed method | Eratosthenes | Sundaram |  |
| 1000 | 15 | 166 | 166 |  | 0.039534 | 0.040474 | 0.040810 |  |
| 10,000 | 50 | 1666 | 1666 |  | 0.045739 | 0.057403 | 0.072236 |  |
| 100,000 | 158 | 16,666 | 16,666 |  | 0.104759 | 0.174620 | 0.214130 |  |
| 1,000,000 | 500 | 166,666 | 166,666 |  | 1.064138 | 1.904619 | 2.361416 |  |
| 10,000,000 | 1581 | 1,666,666 | 1,666,666 |  | 9.957602 | 17.780659 | 22.484089 |  |
| 15,000,000 | 1936 | 2,499,999 | 2,499,999 |  | 16.825560 | 28.163155 | 36.187907 |  |
| 20,000,000 | 2236 | 3,333,332 | 3,333,332 |  | 24.033616 | 39.608600 | 50.809382 |  |
| 25,000,000 | 2500 | 4,166,666 | 4,166,666 |  | 32.062576 | 51.058231 | 66.644372 |  |
| 30,000,000 | 2738 | 4,999,999 | 4,999,999 |  | 44.108035 | 66.276671 | 81.603483 |  |
| 35,000,000 | 2958 | 5,833,332 | 5,833,332 |  | 50.220697 | 77.931476 | 97.107659 |  |
| 40,000,000 | 3162 | 6,666,666 | 6,666,666 |  | 59.943121 | 92.131258 | 116.128039 |  |
| 45,000,000 | 3354 | 7,499,999 | 7,499,999 |  | 70.969666 | 106.166029 | 167.198875 |  |
| 50,000,000 | 3535 | 8,333,332 | 8,333,332 |  | 83.296963 | 120.356633 | 270.250105 |  |
| 55,000,000 | 3708 | 9,166,666 | 9,166,666 |  | 89.159296 | 132.971661 | 446.401534 |  |
| 60,000,000 | 3872 | 9,999,999 | 9,999,999 |  | 108.996244 | 159.264843 | 645.043552 |  |
| 65,000,000 | 4031 | 10,833,332 | 10,833,332 |  | 117.289075 | 166.699335 | 1305.106774 |  |
| 70,000,000 | 4183 | 11,666,666 | 11,666,666 |  | 127.232278 | 187.502824 | 1536.719741 |  |
| 75,000,000 | 4330 | 12,499,999 | 12,499,999 |  | 142.604397 | 201.772238 | 1898.559985 |  |
| 80,000,000 | 4472 | 13,333,332 | 13,333,332 |  | 152.231903 | 223.066406 | 2566.107955 |  |
| 85,000,000 | 4609 | 14,166,666 | 14,166,666 |  | 160.146174 | 322.253137 | 3232.662483 |  |
| 90,000,000 | 4743 | 14,999,999 | 14,999,999 |  | 170.764237 | 410.198350 | 3736.636055 |  |
| 95,000,000 | 4873 | 15,833,332 | 15,833,332 |  | 187.835150 | 753.563263 | 4776.194199 |  |
| 100,000,000 | 5000 | 16,666,666 | 16,666,666 |  | 192.269125 | 1066.539915 | 6802.735193 |  |

104 *A. Elhakeem Abd Elnaby, A.H. El-Baz / Egyptian Informatics Journal 22 (2021) 101–104*

∈

*P*(*m*) = {2} ∪ {2 *r* + 1 : *r* ∈ *N*}

*S*

Also, the cardinality of *P*(*m*) is given by

*S*

*P*(*m*) = *BS* — *AS* ∪ *CS* + 1

*S*

tion. It can be used to generate all primes in order up to a particular number *m N*, *m* P 9. Also; we give an explicit formula which exactly determines the number of primes up to a particular num- ber *m* ∈ *N*, *m* P 9. Moreover, for the sake of comparison, a unified

framework is proposed for obtaining the sieves of both Eratos-

Moreover, [Table 1](#_bookmark15) gives the number of generated sets (k) which

is used to generate the prime numbers and execution times for the three methods versus a particular number *m N*, *m* P 9. It is clear that the proposed method uses less number of generated sets and

∈

less runtimes than the other methods. The results are illustrated in [Fig. 1](#_bookmark14) which indicate the superior performance of our method in obtaining the prime numbers.

The system specifications of the PC that runs the experiments is Intel ® coreTM i7-4790 CPU@ 3.60 GHZ, 16 GB Ram and 16-bit Win- dows operating system, X64-based processor.

1. Conclusion

In this paper, an explicit and direct technique for generating the primes is given. This technique depends on the sets and Floor func-

thenes and Sundaram using our proposed method. The execution times are calculated for the three methods and indicate that the proposed method is more efficient in obtaining the primes in order.

Declaration of Competing Interest

The authors declare that they have no known competing finan- cial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. [Stinson DR. Cryptography: theory and practice. 4th Edition. CRC Press; 2019](http://refhub.elsevier.com/S1110-8665(20)30121-3/h0005).
2. [Crandall R, Pomerance C. Prime numbers: a computational perspective. 2nd](http://refhub.elsevier.com/S1110-8665(20)30121-3/h0010) [Edition. New York: Springer; 2005](http://refhub.elsevier.com/S1110-8665(20)30121-3/h0010).
3. [Cunningham DW. Set theory: a first course. Cambridge University Press; 2016](http://refhub.elsevier.com/S1110-8665(20)30121-3/h0015).