

Creative Commons Attribution 4.0 International (CC BY 4.0)

<https://creativecommons.org/licenses/by/4.0/>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

**Please provide feedback**

Please support the ScholarWorks@UMBC repository by emailing [scholarworks-group@umbc.edu](mailto:scholarworks-group@umbc.edu) and telling us what having access to this work means to you and why it's important to you. Thank you.

# Analytic Proof of the recent Baseline Primality Conjecture

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

07-12-2021 / 12-12-2021

CITATION

Phatak, Dhananjay (2021): Analytic Proof of the recent Baseline Primality Conjecture. TechRxiv. Preprint.  
<https://doi.org/10.36227/techrxiv.17139041.v1>

DOI

[10.36227/techrxiv.17139041.v1](https://doi.org/10.36227/techrxiv.17139041.v1)

# Analytic Proof of the recent Baseline Primality Conjecture

Dhananjay Phatak (phatak@umbc.edu)

CSEE Dept., UMBC, 1000 Hilltop Circle, Baltimore, MD 21250, U.S.A.

2<sup>nd</sup> July 2021, last revised December 6, 2021

## ABSTRACT

This document presents an analytic proof of the Baseline Primality Conjecture (BPC) that was recently unveiled in [1, Part I]. The BPC identifies a new small set of conditions that are sufficient to decide the primality of any input integer  $N$  under test (see [Section 2](#) for the exact statement of the BPC in the original form using algebraic integers; and [Section 3](#) for an equivalent polynomial domain reformulation).

The practical significance of the BPC is that it directly leads to ultra low complexity primality testing algorithms, wherein the number of bit-operations required is essentially a quadratic function of the bit-length of the input  $N$  [1]. More specifically, the Baseline Primality Result (BPR) demonstrates that after an/any integer in the closed interval  $[2, N - 2]$  which is a Quadratic Non Residue (QNR) modulo- $N$  is found; exactly 2 (Two, which is a small  $O(1)$  constant, independent of the bit-length of the input  $N$ ) specific modular exponentiations are sufficient to determine whether  $N$  is a composite or a prime.

The BPC was (and to this day continues to be) extensively tested numerically.<sup>1</sup>

Additionally, analytic proofs of the BPC for several specific forms of the input  $N$  were also presented in [1], wherein the BPR was first unveiled. However, at the time of the original publication [1], we were not able to complete a general analytic proof of the BPC that covered all possible cases (i.e., forms) of the input  $N$ .

We have now completed that vital task by developing a general analytic proof of the BPC using its polynomial domain reformulation. A concise presentation of that analytic proof is the main and narrow focus as well as the main new contribution of this article. An auxiliary contribution is a clear and precise explanation of the intuition behind our approach and the illustration of how it leads to the new theoretical results developed in [1].

---

<sup>1</sup> To date, no counterexample has been found.

## § Section 1 : Brief overview of our recent low complexity Primality Detection Algorithms

In a comprehensive document set recently published in the arXiv archive [1], we unveiled new low complexity deterministic primality testing algorithms. That document set includes three companion manuscripts:

Part I covers the theory behind the methods, followed by specifications of the primality detection algorithms.

Part II (the second companion manuscript) presents extensive experimental (numerical) data to corroborate each of the algorithms and

Part III illustrates proofs of some of the theoretical results underlying the algorithms for some specific forms of the integer  $N$ , under test.

However, a complete analytic proof covering all possible values of the input  $N$  eluded us at the time of the original publication. Consequently, the main results in [1] had to be unveiled as “conjectures”.

The “Baseline Primality Conjecture” is the first major conjecture introduced in [1, Part/Paper I, Section 2]. The BPC states that after a **Quadratic Non Residue (QNR) value** ( $q \bmod N$ ) is found, only two (i.e., a small constant number of) modular exponentiation computations need be performed to decide the primality of  $N$ .

Further, it turns out that any value of  $q$  in the closed interval  $[2, N - 2]$  works; there is no need to find “the smallest QNR modulo  $N$ ” (see [1], and the ensuing Sections in this article for details).

Consequently, after a QNR  $q$  is generated, the number of operations required is upper bounded by  $O\left((\log N)^2 (\text{polylog}(\log N))\right)$ , which is substantially lower than the complexity of other known methods (see “Section 5 in Part I” in [1] for details).

The only caveat is the requirement to generate a QNR value.

To that end, in [1, Part I, Section 5.4], it is demonstrated that it is trivial to generate a QNR modulo  $N$  in all cases except when  $N \bmod 240 = 1$ .

In other words,

an overwhelming majority =  $\frac{119}{120} = 99.16\%$  of all odd integers  $N$  do not need an explicit search for a QNR.

Further, even when a search is needed, fast (low complexity) probabilistic algorithms exist to quickly generate a QNR value [1].

However, as of today, there is no known deterministic polynomial-complexity algorithm to generate a QNR value in every case [1].

**Fortunately, we were able to circumvent that difficulty by developing methods that do not need the explicit numerical value of a quadratic (or higher order) non-residue modulo  $N$  (see Section 8 and onward in part I in [1]).**

**It turns out that any equation whose roots do not exist as modulo- $N$  integers is sufficient ; there is no need to also “solve that equation modulo- $N$ ” and obtain the explicit value of the non-residue. In other words, finding any equation that “implicitly” specifies non-residues as its roots is sufficient.**

**This fact ultimately led to the development of the Generalized Primality Conjectures (GPCs), that in turn led to fully deterministic primality testing algorithms that have the complexity of  $O\left((\log N)^3 (\text{polylog}(\log N))\right)$  bit operations [1].**

***In this article, however, we do not consider those Generalized Primality Conjectures.***

Likewise, we do not address any algorithmic or complexity attributes or explore close connections to other existing primality detection methods (for example, Sections 6 and 7 in Part I in [1] demonstrate close connections between primality tests based on the Baseline Primality conjecture and the well known Miller-Rabin primality test. Fused algorithms that combine best attributes of both methods were also unveiled in [1]).

**Rather, in this article, we narrowly focus only on the Baseline Primality Conjecture (which is the first out of several conjectures developed in [1]) and provide an analytic proof of that result.**

To that end, the next Section reproduces the Baseline Primality Conjecture in the original form [1] using the algebraic-integer  $\sqrt{q}$ .

Then, [Section 3](#) presents an equivalent formulation of the same result in the polynomial domain<sup>2</sup>,

The following section (i.e., [Section 4](#)) presents a few well-known results that are used to prove the main result (i.e., the Baseline Primality Result).

Finally [Section 5](#) demonstrates the proof of the Baseline Primality Conjecture.

The next section (i.e. [Section 6](#)) discusses the relation of our results to the state-of-the-art in Primality Testing and explains the intuition underlying all the new results; including all the results unveiled in [1].

We conclude with brief remarks in [Section 7](#) .

The new contributions of this article are:

[1] The analytic proof of the BPC which is the main focus

and

[2] A clear and precise explanation of the intuition underlying our approach and how it leads to the new theoretical results in [1]; which is an auxiliary contribution.

---

<sup>2</sup> This alternate polynomial domain reformulation also appears in the original document [1] in Part I, Section 3, Remark 3.2 on page number 20.

§ Section 2 : The original form of the conjecture using Algebraic Integers [1]

**Phatak's Baseline Primality Conjecture (PBPC)** : Given any positive integer  $N$  (to be tested for primality), suppose that we find some integer  $q$ , that together with  $N$ , satisfies the following conditions

**C-1 :**  $N > 1$  is an odd integer and is not a perfect square of any other integer.

**C-2 :** The **Jacobi-Symbol** of  $q$  w.r.t.  $N \stackrel{\Delta}{=}^3 \mathbf{Jacobi\_Symbol}(q, N) = -1$   
 $\Rightarrow q$  is a **QNR** (Quadratic Non Residue) modulo- $N$

Note that  $q$  can be any **QNR** value, except one restriction/exclusion:

**C-3 :**  $q \neq -1 \pmod{N}$ .

In other words, only the largest value in  $Z_N^* = (N-1)$  is not acceptable as a **QNR**.

Any other integer in the closed interval  $[2, N-2]$  that is a **QNR** modulo  $N$  works.

**C-4 :**  $q$  satisfies the **Euler Criterion** modulo- $N$

$$q^{\left(\frac{N-1}{2}\right)} \pmod{N} = \mathbf{Jacobi\_Symbol}(q, N) = -1 \pmod{N} . \quad (1)$$

**C-5 :**  $\sqrt{q}$  also satisfies the Modular Binomial Expansion Congruence (**MBEC**) :

$$(1 + \sqrt{q})^N \pmod{N} = 1 + \left[ (\sqrt{q})^N \pmod{N} \right] . \quad (2)$$

**The claim is that if all of the above conditions are satisfied, then  $N$  must be prime.**

---

<sup>3</sup> The symbol  $\stackrel{\Delta}{=}$  denotes a definition.

§ Section 3 : Re-stating the result in terms of polynomial domain remaindering operations

**Phatak's Baseline Primality Theorem (PBPT)** : Given any positive integer  $N$  (to be tested for primality), suppose that we find some integer  $q$ , that together with  $N$ , satisfies the following conditions:

**C-1-p** :  $N > 1$  is an odd integer and is not a square of any other integer.

**C-2-p** : The **Jacobi-Symbol** of  $q$  w.r.t.  $N \triangleq \text{Jacobi\_Symbol}(q, N) = -1$   
 $\Rightarrow q$  is a Quadratic Non Residue (**QNR**) modulo- $N$ .

Note that  $q$  can be any **QNR** value, except one restriction/exclusion:

**C-3-p** :  $q \neq -1 \pmod{N}$ .

In other words, only the largest value  $(N-1)$  in  $Z_N^*$  is not acceptable as a **QNR** ;  
Any other integer in the closed interval  $[2, (N-2)]$  that is a **QNR** modulo- $N$  works.

**C-4-p** :  $q$  satisfies the **Euler Criterion** modulo- $N$ , i.e.,

$$x^{(N-1)} \pmod{\langle (x^2 - q), N \rangle} = \text{Jacobi\_Symbol}(q, N) = -1. \quad (3)$$

**C-5-p** :  $N$  also satisfies the Modular Binomial Expansion Congruence (**MBEC**)

$$(1+x)^N \pmod{\langle (x^2 - q), N \rangle} = 1 + \left[ (x)^N \pmod{\langle (x^2 - q), N \rangle} \right] \quad (4)$$

**The claim is that if all of the above conditions are satisfied, then  $N$  must prime.**

## § Section 4 : Well known results used in the Proof

First, we state a few results that are well known and indicate how to prove some of those results. To that end, we start with some notation and definitions.

Denote

$$\begin{aligned} B(x) &\triangleq \text{the Binomial expansion of } (1+x)^N \\ &= 1 + \binom{N}{1}x + \binom{N}{2}x^2 + \cdots + \binom{N}{k}x^k + \cdots + \binom{N}{N-1}x^{N-1} + x^N \quad . \end{aligned} \quad (5)$$

Likewise, let

$$R(x) \triangleq (1+x)^N - 1 - x^N = B(x) - 1 - x^N \quad (6)$$

$$= \binom{N}{1}x + \binom{N}{2}x^2 + \cdots + \binom{N}{k}x^k + \cdots + \binom{N}{N-1}x^{N-1} \quad . \quad (7)$$

**Fact\_1 : It is well known that [2, 3, 4]:**

$$R(x) \bmod N \begin{cases} = 0 & , \text{ if } N \text{ is prime} \\ \neq 0 & , \text{ if } N \text{ is composite} \end{cases} \quad . \quad (8)$$

**Fact\_2 : When  $N$  is composite, then the terms that survive the remainder operation w.r.t.  $N$  on the right hand side (RHS) of Eqn. (7)**

**are of degrees  $d$  that satisfy :**  $\gcd(d, N) > 1$  . (9)

Some terms with degrees  $d$  satisfying the preceding greatest common divisor (gcd) constraint may not appear,

but if a term does appear, then its degree must satisfy the gcd constraint.

The following two small examples further illustrate the terms that appear on the RHS of Eqn. (7) when  $N$  is a composite number.

**Example 1:**  $N = 35$

$$\left( (1+x)^{35} - 1 - x^{35} \right) \bmod 35 = 7x^5 + 5x^7 + 21x^{10} + 10x^{14} + 10x^{21} + 21x^{25} + 5x^{28} + 7x^{30} \quad . \quad (10)$$

Note that terms of degrees 15 and 20 are not present because

$$\binom{35}{15} \bmod 35 = 0 \quad (11)$$

and

$$\binom{35}{20} \bmod 35 = 0 \quad . \quad (12)$$



**Example 2:**  $N = 12$

$$\left((1+x)^{12} - 1 - x^{12}\right) \bmod 12 = 6x^2 + 4x^3 + 3x^4 + 3x^8 + 4x^9 + 6x^{10} . \quad (13)$$

$$\text{Note that the degree-6 term does not appear because } \binom{12}{6} \bmod 12 = 0 . \quad (14)$$

The next fact-pair directly follows from the previous two facts.

**If  $N$  is a composite number, then**

$$\textbf{Fact\_3\_a} : (B(x) - 1) \bmod N \neq 0 \quad (15)$$

**and**

**Fact\\_3\\_b :**

$$(B(x) - 1) \bmod N \text{ includes } \begin{cases} \text{at least one term with an odd degree of } x \\ \text{and} \\ \text{at least one term with an even degree of } x \end{cases} \quad (16)$$

To prove the last claim for **odd** composite integer  $N$  ,  
let  $p_s$  denote the smallest prime divisor of the composite number  $N$ .  
Then it can be shown that  
the term with the smallest degree (of the argument  $x$ )  
that survives the remaindering operation w.r.t.  $N$  is  $\binom{N}{p_s} x^{p_s}$  ,  
which is an odd-degree term when  $N$  is an odd number.

Using the symmetry of binomial coefficients (i.e., the property  $\binom{N}{k} = \binom{N}{N-k}$ );  
it follows that the term  $\binom{N}{N-p_s} x^{(N-p_s)}$  also is non-zero modulo- $N$  ,  
and it is an even-degree term.  $\square$

This term also happens to be the highest degree term that survives the modulo- $N$  operation.

Next, let

$$B(x) - 1 = (1+x)^N - 1 \triangleq x P(x), \quad \text{so that} \quad (17)$$

$$\begin{aligned} P(x) &\triangleq \frac{B(x) - 1}{x} \\ &= \binom{N}{1} + \binom{N}{2} \cdot x + \cdots + \binom{N}{k} \cdot x^{(k-1)} + \cdots + \binom{N}{N-1} x^{(N-2)} + x^{(N-1)} \end{aligned} \quad (18)$$

**and**

$$R(x) = x P(x) - x^N . \quad (19)$$

From **Fact\\_3**, it follows that

**Fact\\_4 : If  $N$  is a composite number, then**

$$P(x) \bmod N \neq 0 \quad (20)$$

**and**

$$(P(x) \bmod N) \text{ includes } \begin{cases} \text{at least one term with an odd degree of } x \\ \text{and} \\ \text{at least one term with an even degree of } x . \end{cases} \quad (21)$$

For instance, in Example 1, where  $N = 35$ ,

$$(B(x) - 1) \bmod N = 7x^5 + 5x^7 + 21x^{10} + 10x^{14} + 10x^{21} + 21x^{25} + 5x^{28} + 7x^{30} + x^{35} \quad (22)$$

and

$$P(x) \bmod N = 7x^4 + 5x^6 + 21x^9 + 10x^{13} + 10x^{20} + 21x^{24} + 5x^{27} + 7x^{29} + x^{34} \quad (23)$$

Likewise, in Example 2, where  $N = 12$ ,

$$(B(x) - 1) \bmod N = 6x^2 + 4x^3 + 3x^4 + 3x^8 + 4x^9 + 6x^{10} + x^{12} \quad (24)$$

and

$$P(x) \bmod N = 6x + 4x^2 + 3x^3 + 3x^7 + 4x^8 + 6x^9 + x^{11} \quad (25)$$

Next, note that any arbitrary polynomial  $\mathcal{L}(x)$  can always be written as the sum of its even and odd parts:

$$\mathcal{L}(x) = \mathcal{L}_{\text{even}}(x) + \mathcal{L}_{\text{odd}}(x) \quad , \quad (26)$$

where ,

$$\mathcal{L}_{\text{even}}(x) \triangleq \text{sum of all even degree terms} \Rightarrow \mathcal{L}_{\text{even}}(-x) = \mathcal{L}_{\text{even}}(x) \quad (27)$$

and

$$\mathcal{L}_{\text{odd}}(x) \triangleq \text{sum of all odd degree terms} \Rightarrow \mathcal{L}_{\text{odd}}(-x) = -\mathcal{L}_{\text{odd}}(x) \quad (28)$$

Also note that

$$\mathcal{L}_{\text{even}}(x) = \frac{1}{2}(\mathcal{L}(x) + \mathcal{L}(-x)) \quad (29)$$

and

$$\mathcal{L}_{\text{odd}}(x) = \frac{1}{2}(\mathcal{L}(x) - \mathcal{L}(-x)) \quad (30)$$

### **Even Polynomial Lemma 1:**

**If  $\mathcal{L}(x) = \mathcal{L}_{\text{even}}(x)$  is an even polynomial,**

**then  $\mathcal{L}(x) \bmod \langle (x^2 - q), N \rangle = c_0$  (i.e., a constant-term or a 0-degree term).**

**Proof :** Let

$$\mathcal{L}(x) = \mathcal{L}_{\text{even}}(x) = \sum_{i=1}^n C_{2d_i} \cdot (x)^{2d_i} \quad (31)$$

Then

$$\mathcal{L}(x) \bmod \langle (x^2 - q), N \rangle = \sum_{i=1}^n C_{2d_i} \cdot \left( x^{2d_i} \bmod \langle (x^2 - q), N \rangle \right) \quad (32)$$

$$= \sum_{i=1}^n C_{2d_i} \left[ \left( x^2 \bmod (x^2 - q) \right)^{d_i} \bmod N \right] \quad (33)$$

Plug-in  $x^2 \bmod (x^2 - q) = q$  in the preceding equation to obtain

$$\begin{aligned} \mathcal{L}_{\text{even}}(x) \bmod \langle (x^2 - q), N \rangle &= \sum_{i=1}^n C_{2d_i} [(q)^{d_i} \bmod N] \\ &= \left( \sum (\text{constant or 0-degree terms}) \right) \bmod N \text{ which is a constant} \quad \square \end{aligned} \quad (34)$$

**Odd\_Polynomial\_Lemma\_2:**

**If  $\mathcal{L}(x) = \mathcal{L}_{\text{odd}}(x)$  is an odd polynomial,**

**then  $\mathcal{L}(x) \bmod \langle (x^2 - q), N \rangle = c_1 x$ , where  $c_1$  is a modulo- $N$  integer.**

**Proof :** Note that

$$\mathcal{L}_{\text{odd}}(x) = x \mathcal{L}_{\text{even}}(x) \quad (35)$$

and apply **Even\_Polynomial\_Lemma\_1** to obtain the desired result.  $\square$

Finally, we state a fifth fact, which directly follows from **Fact\_4**.

**Fact\_5 :**

$$\left. \begin{aligned} (P(x) + P(-x)) \bmod N &\neq 0 \\ \text{and} \\ (P(x) - P(-x)) \bmod N &\neq 0 \end{aligned} \right\} \quad (36)$$

**in other words ,**

$$(P(x) \bmod N) \text{ is neither even nor odd.}$$

## § Section 5 : Proof of the Baseline Primality Theorem

### § Subsection 5.1 : Explanation of the exclusion of $q = -1 \pmod N$ as the QNR

First, we briefly explain why the QNR value  $q = -1$  must be excluded.

The Euler's Criterion Check (condition [C-4-p](#) from [Section 5](#)), when expressed via polynomial remainder operation, takes the form

$$x^{(N-1)} \pmod{\langle (x^2 - q), N \rangle} = -1 \quad . \quad \text{which is Eqn. (3) repeated for convenience}$$

The preceding equation can be re-arranged as

$$(x^{(N-1)} + 1) \pmod{\langle (x^2 - q), N \rangle} = 0 \quad . \quad (37)$$

Since  $N$  is odd,  $(N - 1)$  is an even integer. Accordingly,

$$\text{let } \frac{N-1}{2} = \Delta \quad . \quad (38)$$

Then, Eqn. (37) can be expressed in terms of  $\Delta$  as follows:

$$((x^2)^\Delta + 1) \pmod{\langle (x^2 - q), N \rangle} = 0 \quad . \quad (39)$$

If QNR  $q = -1$  is used, then the divisor polynomial is

$$\mathcal{D}(x) = x^2 - q = x^2 - (-1) = x^2 + 1 \quad (40)$$

and Eqn. (39) becomes

$$((x^2)^\Delta + 1) \pmod{\langle (x^2 + 1), N \rangle} = 0 \quad . \quad (41)$$

Note that

$$\text{if } \Delta \text{ is an odd number, then } ((x^2)^\Delta + 1) \text{ is divisible by } (x^2 + 1) \quad . \quad (42)$$

**irrespective of any other condition on  $N$ .**

As a result, a large number of arbitrary values of  $N$  (that are not prime numbers or Carmichael numbers) can satisfy the Euler's Criterion, if  $q = -1$  is allowed.

To steer clear of this obstacle, we simply disallow  $q = -1$ , so that

$$(x^2 - q) \pmod N \neq x^2 + 1 \quad . \quad (43)$$

Consequently, the full discriminating power of the Euler Criterion Check is leveraged.

Next, we proceed with the rest of the proof.

## § Subsection 5.2 : Main body of the Proof

Condition **(C-5-p)** in the theorem requires us to explicitly check that the following identity holds with  $x$  as a variable/indeterminate:

$$R(x) \bmod \langle (x^2 - q), N \rangle \stackrel{\checkmark}{=} 0 . \quad (44)$$

Note that the remainder w.r.t. a quadratic divisor of the form  $(x^2 - q)$  is at most a degree 1 polynomial. Accordingly, we evaluate remainders modulo  $(x^2 - q)$  and modulo  $N$  in each square-and-reduce step in each modular exponentiation and verify that the final remainder is indeed 0. (a polynomial 0  $\Rightarrow$  coefficients of all degrees in the remainder polynomial are equal to  $(0 \bmod N)$ ) .

Therefore, the preceding equation implies that one of the following two cases must hold:

**Case 1:**  $R(x) \bmod N = 0$  ; i.e., the entire polynomial  $[R(x) \bmod N] = 0$  .  
as a result , taking the remainder of 0 w.r.t. divisor polynomial  $\mathcal{D}(x)$  also yields 0 as per Eqn. (44) .  
In this case, equating coefficients of each degree of  $x$  on both sides of Eqn. (7) yields

$$({}^N C_k) \bmod N = 0 \quad \text{for all } 1 \leq k \leq (N-1) . \quad (45)$$

The preceding relations imply that  $N$  must be prime.  $\square$

or

$$\textbf{case 2: } [R(x) \bmod N] \neq 0 \text{ but } [(R(x) \bmod N) \bmod (x^2 - q)] = 0 \quad (46)$$

We show that this case, together with the Euler-Criterion-Check (condition **C-4-p**), leads to a contradiction.

To enhance the readability, we split the rest of the proof into 3 Steps:

**Step\_1: Derive the canonical identity<sup>4</sup> i.e., Eqn. (65) .**

To that end, we first re-state the Euler Criterion Check condition for the sake of convenience and clarity:

$$x^{(N-1)} \bmod \langle (x^2 - q), N \rangle = \mathbf{Jacobi\_Symbol}(q, N) = -1 \quad .$$

Then, assuming that  $x \neq 0$ , multiply both sides of the preceding equation by  $x$  to obtain

$$x^N \bmod \langle (x^2 - q), N \rangle = -x \quad . \quad (47)$$

Next, plug-in “ $-x$ ” for “ $x^N$ ” on the right hand side of condition [C-5-p](#) to obtain

$$(1+x)^N \bmod \langle (x^2 - q), N \rangle = 1 + x^N = (1 - x) \quad (48)$$

$$\Rightarrow (1+x)^N - 1 + x = 0 \bmod \langle (x^2 - q), N \rangle \quad . \quad (49)$$

From the definition of  $P(x)$  in Eqn. (17), the preceding equation can be re-written as

$$xP(x) + x = x(1 + P(x)) \bmod \langle (x^2 - q), N \rangle = 0 \quad , \quad (50)$$

which, after dividing both sides by  $x$  yields

$$(1 + P(x)) \bmod \langle (x^2 - q), N \rangle = 0 \quad (51)$$

$$\Rightarrow P(x) \bmod \langle (x^2 - q), N \rangle = -1 \quad . \quad (52)$$

Next, we re-write Eqn. (17) for convenience:

$$B(x) - 1 = (1+x)^N - 1 \triangleq xP(x) \quad .$$

Differentiate both sides of the preceding equation w.r.t. argument  $x$  to obtain

$$\frac{d}{dx}(xP(x)) = \frac{d}{dx}((1+x)^N - 1) = N(1+x)^{(N-1)} \quad . \quad (53)$$

The preceding relation directly follows from the Binomial Theorem and is valid for arbitrary scalar values of the argument  $x$ . In other words, the preceding relation holds for real values of  $x$ . Taking the derivative w.r.t.  $x$  is therefore a valid operation.

Using the standard short-hand notation; wherein the superscript “ $\{ '\}$ ” (i.e. the symbol “dash”) denotes a derivative, i.e.,

$$\frac{d}{dx}[F(x)] \triangleq F'(x) \quad , \quad (54)$$

---

<sup>4</sup> Identifying/labeling Eqn. (65) as **the canonical identity** and the subsequent Eqn. (73) as its **conjugate identity** allows a clean partitioning of the sequence of arguments into three logical steps.

In other words, those labels are only intended to indicate logical step boundaries and should not be misinterpreted to imbue those two equations with any other significance; since every equation included in the proof is necessary; and is therefore no less important than any other equation.

Eqn. (53) can be written as

$$[xP(x)]' = N(1+x)^{(N-1)} \quad . \quad (55)$$

Take the remainder of both sides of the preceding equation w.r.t.  $N$  to obtain

$$[xP(x)]' \bmod N = [N(1+x)^{(N-1)}] \bmod N = 0 \bmod N \quad , \quad (56)$$

or equivalently,

$$xP'(x) + P(x) = 0 \bmod N \quad (57)$$

Reduce the preceding equation modulo  $(x^2 - q)$  to obtain

$$[x \cdot P'(x) + P(x)] \bmod \langle (x^2 - q), N \rangle = 0 \quad (58)$$

$$\Rightarrow x \cdot P'(x) = -P(x) \bmod \langle (x^2 - q), N \rangle \quad . \quad (59)$$

Then, from Eqn. (52), plug-in  $P(x) \bmod \langle (x^2 - q), N \rangle = -1$  in the preceding equation to obtain

$$xP'(x) \bmod \langle (x^2 - q), N \rangle = -(-1) = 1 \quad . \quad (60)$$

The preceding equation implies that

$$\gcd(P'(x), (x^2 - q)) = 1 \quad (61)$$

and

$$\gcd(P'(x), N) = 1 \quad . \quad (62)$$

Because if any of the two preceding “gcd” values were to be bigger than 1, then the right hand side of Eqn. (60) would have included each of those gcd values as a factor, and therefore the RHS would not be equal to 1.

The preceding equation implies that

$$\left( \frac{1}{P'(x)} \right) \bmod N \quad \text{exists and is unique} \quad . \quad (63)$$

$$\Rightarrow \text{Multiplication and division by } P'(x) \text{ are valid operations modulo } N \quad . \quad (64)$$

Consequently, dividing both sides of Eqn. (57) by  $P'(x)$  and re-arranging yields

$$\text{the canonical identity:} \quad x = \frac{-P(x)}{P'(x)} \bmod N \quad . \quad (65)$$

At this point Step\_1 is complete, which brings us to the next step:

**Step\_2: Derive the conjugate canonical identity ( Eqn. (73) )**

Next, we derive a set of analogous equations , where , the argument of the polynomials is “ $(-x)$ ”.

To that end, start by replacing “ $x$ ” with “ $-x$ ” in Eqn. (17) to obtain

$$B(-x) - 1 = (1 - x)^N - 1 = (-x) P(-x) . \quad (66)$$

Take the derivative of the preceding equation w.r.t.  $x$  to obtain

$$[(-x) \cdot P(-x)]' = [(1 - x)^N - 1]' = N \times (1 - x)^{(N-1)} \times (-1) . \quad (67)$$

Take the remainder of the preceding equation w.r.t.  $N$  to obtain

$$[(-x) \cdot P(-x)]' = 0 \pmod{N} \quad (68)$$

$$\Rightarrow [(-1) \times x \times P(-x)]' = 0 \pmod{N} \Rightarrow (-1) \times [x \times P(-x)]' = 0 \pmod{N}$$

$$\Rightarrow [x \times P(-x)]' = 0 \pmod{N} \quad (69)$$

which results in

$$x \times P'(-x) \times (-1) + P(-x) = 0 \pmod{N} . \quad (70)$$

Then, using the same arguments that were used to establish the validity of modulo  $N$  multiplication and division by  $P'(x)$  , it follows that

$$\left( \frac{1}{P'(-x)} \right) \pmod{N} \quad \text{also exists and is unique} . \quad (71)$$

$$\Rightarrow \text{Multiplication and division by } P'(-x) \text{ are also valid operations modulo-}N . \quad (72)$$

Therefore, dividing Eqn. (70) by  $P'(-x)$  and re-arranging yields

$$\text{the conjugate canonical identity: } x = \frac{P(-x)}{P'(-x)} \pmod{N} , \quad (73)$$

which brings us to the end of Step\_2 and leads to the final step:

---



### Step\_3: Demonstrate that the canonical identity and its conjugate lead to a contradiction

Finally, from Eqn. (65) and Eqn. (73) we obtain

$$x = \frac{-P(x)}{P'(x)} \pmod{N} = \frac{P(-x)}{P'(-x)} \pmod{N} , \quad (74)$$

which implies that

$$-\left(\frac{P(x)}{P'(x)}\right) = \left(\frac{P(-x)}{P'(-x)}\right) \pmod{N} . \quad (75)$$

For clarity let

$$H(x) \triangleq \frac{P(x)}{P'(x)} \pmod{N} . \quad (76)$$

Then, Eqn. (75) can be re-written as

$$H(-x) = -H(x) \pmod{N} , \quad (77)$$

**which implies that  $H(x)$  is an odd function.** (78)

There are two possible paths to render  $H(x)$  into an odd function:

**Path\_1:**  $P(x)$  is odd  $\Rightarrow P'(x)$  is even ,

because the degree of each term in  $P'(x)$  is one less than the corresponding term in  $P(x)$ .

**Therefore the ratio  $\frac{P(x)}{P'(x)} = H(x)$  is an odd function.**

or

**Path\_2:**  $P(x)$  is even  $\Rightarrow P'(x)$  is odd ,

**therefore their ratio  $H(x)$  is again an odd function.**

**However, note that Eqn. (52) and the Odd\_Polynomial\_Lemma\_2 rule out Path\_1:**

(because if we assume that  $P(x)$  is odd, then as per Odd\_Polynomial\_Lemma\_2, its reduction modulo  $\langle (x^2 - q), N \rangle$  is of the form  $(c_1x)$  or a degree-1 term.

Eqn. (52), however, indicates that the reduction of  $P(x)$  modulo  $\langle (x^2 - q), N \rangle$  is a degree-0 term. Therefore by the contrapositive of Odd\_Polynomial\_Lemma\_2, the hypothesis that  $P(x)$  is odd is incorrect.)

**Therefore we are left with Path\_2 ,**

**which implies that  $(P(x) \pmod{N})$  is an even polynomial, which contradicts Fact\_5, and completes the proof.  $\square$**

## § Section 6 : Discussion: our results advance the state-of-the-art of Primality Testing

In this section, we first present a brief summary of the current state-of-the-art of deterministic primality testing. We then explain the intuition behind our methods<sup>5</sup> and demonstrate how it leads to new results; including all of the results unveiled in [1].

### § Subsection 6.1 : Brief summary of current state-of-the-art of Deterministic Primality Testing

It is well known [2] that, any integer  $N$  is a prime number **iff** the following Modular Binomial Expansion Congruence (**MBEC**) holds

$$\mathbf{MBEC} \quad : \quad (x + y)^N \bmod N = (x^N + y^N) \bmod N, \quad (79)$$

where,  $x$  and  $y$  are arbitrary scalar integers or scalar indeterminates/variables.

The proof is based on another well known fact [2, 3, 4]: For every integer  $N > 1$ ,

$$({}^N C_k) \bmod N = 0 \quad \text{for } k = 1, \dots, (N-1) \quad \mathbf{iff} \quad N \text{ is a prime, where} \quad (80)$$

$$({}^N C_k) = \frac{N \cdot (N-1) \cdots (N-k+1)}{1 \cdot 2 \cdots k} = \text{the binomial coefficient } N\text{-choose-}k. \quad (81)$$

For a partial and intuitive explanation of why Eqn. (80) holds: note that if  $N$  is a prime number, then none of the factors in the denominator of the binomial coefficient divides  $N$ ; and this is true for all non-trivial binomial coefficients, i.e., for  $k$  values satisfying  $1 \leq k \leq N-1$ . As a result, if  $N$  is a prime, then  $N$  divides all non-trivial binomial coefficients; which in turn implies that the remainder of any non-trivial binomial coefficient modulo  $N$  is zero.

Purely symbolic direct verification of the **MBEC** (i.e., Eqn. (79)) leaving  $x$  and  $y$  as true indeterminate symbols is not possible for all but small toy values of  $N$  (because the number of terms in the Binomial Expansion is  $N+1$ ).

The renowned AKS method [2] uses a slightly restricted form of Eqn. (79) wherein they leave a single variable argument  $x$  and select some specific integer value “ $a$ ” to be substituted in place of the second variable  $y$ , resulting in the congruence

$$(x + a)^N \bmod N = (x^N \bmod N) + (a^N \bmod N). \quad (82)$$

---

<sup>5</sup> in prior versions of this document, a sizable fraction of the material presented in this section was placed right after the Introduction. However, some readers commented that with that arrangement, the statement of the Baseline Primality Result got deferred all the way to page number 8; and the main body of the proof did not start until page 14, thereby locating the proof in the last third of that document; which seemed to deviate from the intention and claim to narrowly focus the document on the analytic proof of the BPR. Therefore in this version, we have kept the introduction to the bare minimum (approximately 2 pages long) and delved into the statement of the Baseline Primality Result and its proof at the earliest possible juncture. We hope that this re-organization keeps the main focus of this document on the BPR and its proof as intended.

By Fermat's Little Theorem, if  $N$  is any prime number, then  $a^N \bmod N = a$ , for every integer  $a \in [1, N-1]$ , which further simplifies the preceding equation to

$$(x+a)^N \bmod N = (x^N \bmod N) + a. \quad (83)$$

Even with this simplification, direct verification of the preceding equation leaving the argument  $x$  as a true indeterminate symbol is not possible for all but small toy values of  $N$ .

To circumvent this difficulty the AKS method and its derivatives or variants [3, 5, 6, 7, 8, 9], in essence, can be thought to invoke the following work-around: suppose that instead of trying to verify Eqn. (83) directly; we take the remainder of the identity in that equation with respect to some divisor polynomial  $\mathcal{D}(x)$  to obtain the the following modified equation to be tested:

$$[(a+x)^N \bmod N] \bmod \mathcal{D}(x) = [(a+x^N) \bmod N] \bmod \mathcal{D}(x), \quad (84)$$

or equivalently

$$\left[ \left( (a+x)^N - a - x^N \right) \bmod N \right] \bmod \mathcal{D}(x) = 0. \quad (85)$$

If the degree  $\mathcal{D}(x)$  is sufficiently small, then it is possible to check the congruence in the preceding equation in a computationally efficient manner while letting  $x$  remain a true indeterminate/variable.

However, the main question then becomes: can the divisor polynomial be selected in such a way that whenever the congruence in Eqn. (85) holds,

the original congruence in Eqn. (83) that we would like to verify/check also holds true?

How many such divisor polynomials need to be tried ?

At how many distinct “ $a$ ” values does the congruence in Eqn. (83) need to be tested (in order to guarantee that it is true for all integer values of  $x$  and  $a$ ) ?

The AKS family of methods deploy a divisor polynomial of the form

$$\mathcal{D}(x) = x^r - 1. \quad (86)$$

The ingenuity of the AKS deterministic primality test [2] lies in demonstrating that the congruence in Eqn. (83) holds for all integer values of  $x$  as long as

(i) the degree  $r$  of the divisor polynomial  $\mathcal{D}(x)$  satisfies a logarithmic bound, w.r.t.  $N$ ;

and

(ii) the modular congruence in Eqn. (84) is checked at all integer values of “ $a$ ” up to some threshold value, which is also logarithmically bounded.

## § Subsection 6.2 : Motivation behind our recent work

We took a slightly different approach, considering a more specific form of the MBEC

$$\text{Specific version of MBEC (SvMBEC)} : (1+x)^N \bmod N = (1+x^N) \bmod N, \quad (87)$$

where,  $x$  is a scalar indeterminate.

If the preceding **SvMBEC** (i.e., Eqn. (87)) could be verified at some numerical value of  $x$  such that none of the powers of that numerical value exist as integers, then that single verification would be sufficient to conclude that  $N$  must be prime.

For example, if the **SvMBEC** could be verified at a single transcendental real value of  $x$  (such as,  $x = \pi^6$ ; or  $x = e$  = the base of natural logarithms; or any real number that is not an algebraic integer); then that single verification should be sufficient to conclude that  $N$  must be a prime number.

However, it is not immediately clear how to compute sufficiently accurate floating-point approximations of the “remainders” that would arise in such a numerical verification, in an efficient manner. It is likely that the precision (and consequently the total amount of computations) required to verify the **SvMBEC** at transcendental real values of  $x$  is impractically large for all but small toy values of  $N$ .

Consequently, like the AKS method and its variants, we also test the preceding **SvMBEC** congruence modulo some divisor polynomial  $\langle \mathcal{D}(x), N \rangle$ :

$$(1+x)^N \bmod \langle \mathcal{D}(x), N \rangle = 1 + (x^N) \bmod \langle \mathcal{D}(x), N \rangle, \quad (88)$$

where,

1. (and throughout this manuscript) “ $x$ ” denotes the indeterminate/variable argument of all polynomials.
2. It is assumed that all coefficients of any polynomial are modulo- $N$  integers; or in other words every coefficient in any polynomial is an integer in the closed interval  $[0, (N-1)]$  .  
and
3. The notation  $\mathcal{P}(x) \bmod \langle \mathcal{D}(x), N \rangle$  denotes the result obtained by
  - (i) reducing (i.e., taking the remainder of) polynomial  $\mathcal{P}(x)$  with respect to the divisor polynomial  $\mathcal{D}(x)$ ,
  - (ii) and then reducing all the coefficients in the polynomial remainder modulo- $N$ .

It turns out that the order of the remaindering operations (w.r.t.  $\mathcal{D}(x)$  and  $N$ ) does not affect the final result.

---

<sup>6</sup> “Archimedes’ ” constant = 3.1415...

**The substantial difference between the AKS family of methods and our methods lies in how we select the divisor polynomial  $\mathcal{D}(x)$ :**

Unlike AKS, instead of focusing on a specific form of the polynomial, we deploy divisor polynomials  $\mathcal{D}(x)$  such that

$$\mathcal{D}(x) = 0 \pmod{N} \text{ has no integer roots} . \quad (89)$$

As a result, verifying the congruence in Eqn. (88) is tantamount to verifying the main congruence of interest in Eqn. (87) at the roots of  $\mathcal{D}(x) = 0$ ,  
**i.e., at values that do not exist as modulo- $N$  integers.**

To see this fact, suppose that

$$(1+x)^N \pmod{N} = (Q_1(x) \cdot \mathcal{D}(x) + R_1(x)) \pmod{N} \quad (90)$$

and

$$1+(x^N) \pmod{N} = (Q_2(x) \cdot \mathcal{D}(x) + R_2(x)) \pmod{N} , \quad (91)$$

where,  $Q_1(x)$ ,  $Q_2(x)$  are the quotients and  $R_1(x)$ ,  $R_2(x)$  are the corresponding remainders.

Then, evaluate Eqn. (90) and Eqn. (91) at  $x = \text{root\_of\_}\mathcal{D}(x)$  and note that

$$\left| \mathcal{D}(x) \right|_{x = \text{root\_of\_}\mathcal{D}(x)} = 0 . \quad (92)$$

The left hand side of the preceding equation has the standard notation for the evaluation of an expression in between the vertical lines, at the argument value, which is indicated in the subscript of the second vertical delimiter.

Then, it follows that

$$\left( \left| R_1(x) \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} = \left( \left| (1+x)^N \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} \quad (93)$$

and

$$\left( \left| R_2(x) \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} = \left( \left| 1+(x)^N \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} . \quad (94)$$

Further, if Congruence (88) is satisfied , then the remainders are equal, i.e.,

$$R_1(x) = R_2(x) \pmod{\langle \mathcal{D}(x), N \rangle} \quad (95)$$

$$\Rightarrow \left( \left| R_1(x) \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} = \left( \left| R_2(x) \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} , \quad (96)$$

which together with Eqn. (93) and Eqn. (94) , yields

$$\left( \left| (1+x)^N \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} = \left( \left| 1+(x)^N \right|_{x = \text{root\_of\_}\mathcal{D}(x)} \right) \pmod{N} . \quad (97)$$

The preceding three equations demonstrate that verifying the congruence in Eqn. (88) is tantamount to verifying the main congruence of interest in Eqn. (87), at the roots of  $\mathcal{D}(x) = 0$ .

**We deliberately select  $\mathcal{D}(x)$  such that it has no integer roots modulo- $N$ .**

**Therefore, the roots of  $\mathcal{D}(x)$  are algebraic integers including irrational real numbers that do not exist as modulo- $N$  integers.**

**As a result, our selection of  $\mathcal{D}(x)$  enables an implicit verification of the congruence in Eqn. (87) at irrational real values of the argument  $x$  that do not exist as modulo- $N$  integers.**

★ **Therefore one single check of this type should suffice to conclude that  $N$  is prime.** ★

A sub case of the preceding argument was formalized as the “Baseline Primality Conjecture” and has been extensively investigated in [1]; and analytically proved in this document.

## § Section 7 : Concluding Remarks

We have demonstrated an analytic proof of the recent Baseline Primality Conjecture introduced in [1]. The main highlights of the proof are:

- It is an elementary proof by contradiction.
- A striking and intriguing attribute of the proof is that it requires an analysis of only the odd and even symmetry properties of some polynomials.  
In other words, the proof does not depend on the exact form of the polynomials (i.e., the exact values of the degrees of the terms in those polynomials or their coefficients), which in turn suggests that similar arguments might work to prove the Generalized Conjectures unveiled in [1].
- (We think) that it is concise as well as clear.

The underlying motivation brings out the benefits of pushing the boundaries to see whether fundamental results (such as the binomial expansion theorem) yield additional insights or complexity advantages when applied in domains different from the original ones: for example, letting the variables in the binomial theorem be real numbers or even Matrices (satisfying the added constraint that the product of the two matrices must be commutative). Indeed the matrix experiments in testing primality was how this entire line of investigation originally started in the first place (see [1] for further details).

The practical significance of the results presented in this article is substantial because the analytic proof of the underlying fundamental theoretical result is the first and most important component of any larger (or more global) proof of correctness of any primality testing algorithms that are based on that theoretical result.

At the time of this writing, other best known deterministic primality testing algorithms, such as [3, 5, 6, 7, 8, 9], have a complexity that is quartic or higher (depending on the details of how the elementary arithmetic operations are defined and their time-delay, memory and power requirements on a typical processor are characterized). It turns out that for cryptographically secure integer lengths of 1024-bits (or higher); the other known deterministic algorithms are still too slow (despite having a polynomial complexity). As a result, all real-world implementations (including the GNU as well as Python unlimited precision libraries, Maple, etc) deploy few iterations of the Miller-Rabin test (together with few other tests). In other words, all real life implementations still use only probabilistic primality testing methods. While this has not caused a serious problem as of today, it is desirable to replace the probabilistic methods with deterministic methods. The algorithms based on the BPT that are unveiled in [1] represent a big step in that direction.

Now that the Baseline Primality Result has been analytically proved; we enthusiastically invite readers, reviewers and peers to take the next logical step, which is to prove<sup>7</sup> (or in the unlikely worst case scenario, disprove) the other conjectures unveiled in [1].

## ACKNOWLEDGMENTS

The author would like to thank his colleague Professor Alan T. Sherman for his prompt, detailed and helpful comments on multiple versions of this document.

The author also thanks Prof. Erich Bach from the CS Dept. at the University of Wisconsin, Madison for providing proofs of some critical conjectures (other than the BPC) that were unveiled in [1]. See footnote number 6 at the bottom of page number 20, and reference [10] for further details.

---

<sup>7</sup>For instance, on 31<sup>st</sup> August 2019 ; Prof. Eric Bach, our colleague from CS Dept., Univ of Wisconsin; emailed us the proofs of two other auxiliary conjectures unveiled in the same document [1].

In particular he has provided the proofs of Equations numbered (78) and (79) in [1] .  
He was extremely generous and told us that we could use the PDF document he provided us or parts thereof anywhere as and when needed. We have therefore made his proofs available via the UMBC CSEE Dept. home page of the author of this article. Please see reference [10] in the bibliography for further details.



# Bibliography

- [1] D. S. Phatak, et. al., “PPT : New Low Complexity Deterministic Primality Tests Leveraging Explicit and Implicit Non-Residues.”  
August 2019,  
The overall document is a set of 3 companion articles available via the following arXiv url.  
[Online]. Available: <https://arxiv.org/abs/1908.06964>
- [2] M. Agrawal, N. Kayal, and N. Saxena, “PRIMES is in P,” *Annals of mathematics*, pp. 781–793, 2004.
- [3] R. Crandall and C. B. Pomerance, *Prime numbers: a computational perspective*. Springer Science & Business Media, 2006, vol. 182.
- [4] V. Shoup, *A computational introduction to number theory and algebra*. Cambridge university press, 2009.
- [5] M. Dietzfelbinger, *Primality testing in polynomial time: from randomized algorithms to "PRIMES is in P"*. Springer, 2004, vol. 3000.
- [6] H. W. Lenstra Jr. and C. Pomerance, “Primality testing with gaussian periods,” in *FSTTCS*, 2002, p. 1.
- [7] H. Lenstra Jr. and C. Pomerance, “Primality testing with gaussian periods,” Last modified : 2008, . [Online]. Available: <https://math.dartmouth.edu/~carlp/aks240817.pdf>
- [8] D. Bernstein, “Proving primality in essentially quartic random time,” *Mathematics of computation*, vol. 76, no. 257, pp. 389–403, 2007.
- [9] H. W. Lenstra Jr and C. B. Pomerance, “Primality testing with gaussian periods,” *Journal of the European Mathematical Society*, vol. 21, no. 4, pp. 1229–1269, 2019 . [Online]. Available: <https://math.dartmouth.edu/~carlp/aksfinal.pdf>
- [10] **Bach, Eric.**, “Personal communication : Proofs of Auxiliary\_Conjecture\_1 Identities numbered as Equations (78) and (79) in [1],” Computer Science Dept., Univ. of Wisconsin, Madison, PDF document received via Email on the 31<sup>st</sup> August 2019  
. [Online]. Available: <http://www.csee.umbc.edu/~phatak/newres/dissemin/Eric-Bach-proofs-of-eqns-78-79-31aug2019.pdf>