Journal of Applied Mathematics and Physics, 2019, 7, 364-380
http://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

How to Check If a Number Is Prime Using a Finite Definite Integral

Jesús Sánchez

Independent Researcher, Bilbao, Spain
Email: jesus.sanchez.bilbao@gmail.com

How to cite this paper: Sánchez, J. (2019) How to Check If a Number Is Prime Using a Finite Definite Integral. *Journal of Applied Mathematics and Physics*, **7**, 364-380.
https://doi.org/10.4236/jamp.2019.72028

Received: January 11, 2019
Accepted: February 19, 2019
Published: February 22, 2019

Abstract

In the history of mathematics different methods have been used to detect if a number is prime or not. In this paper a new one will be shown. It will be demonstrated that if the following equation is zero for a certain number p, this number p would be prime. And being m an integer number higher than $(p+1)/2$ (the lowest, the most efficient the operation).

$\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{pj\omega}\left(\sum_{k=2}^{k=m+1} e^{-2kj\omega}\left(\frac{1-e^{-mkj\omega}}{1-e^{-kj\omega}}\right)\right)d\omega$. If the result is an integer, this result will tell us how many permutations of two divisors, the input number has. As you can check, no recurrent division by odd or prime numbers is done, to check if the number is prime or has divisors. To get to this point, we will do the following. First, we will create a domain with all the composite numbers. This is easy, as you can just multiply one by one all the integers (greater or equal than 2) in that domain. So, you will get all the composite numbers (not getting any prime) in that domain. Then, we will use the Fourier transform to change from this original domain (called discrete time domain in this regards) to the frequency domain. There, we can check, using Parseval's theorem, if a certain number is there or not. The use of Parseval's theorem leads to the above integral. If the number p that we want to check is not in the domain, the result of the integral is zero and the number is a prime. If instead, the result is an integer, this integer will tell us how many permutations of two divisors the number p has. And, in consequence information how many factors, the number p has. So, for any number p lower than $2m - 1$, you can check if it is prime or not, just making the numerical definite integration. We will apply this integral in a computer program to check the efficiency of the operation. We will check, if no further developments are done, the numerical integration is inefficient computing-wise compared with brute-force checking. To be added, is the question regarding the level of accuracy needed (number of decimals and number of steps in the numerical integration) to

have a reliable result for large numbers. This will be commented on the paper, but a separate study will be needed to have detailed conclusions. Of course, the best would be that in the future, an analytical result (or at least an approximation) for the summation or for the integration is achieved.

Keywords

Primality Test, Number Theory, Primes, Factorization, Fourier Transform, Parseval's Theorem, Time Domain, Frequency Domain, Numerical Computation

1. Introduction

In this paper, we will obtain the following integral:

$$\frac{1}{2\pi}\int_{-\pi}^{\pi} e^{pj\omega}\left(\sum_{k=2}^{k=m+1} e^{-2kj\omega}\left(\frac{1-e^{-mkj\omega}}{1-e^{-kj\omega}}\right)\right)d\omega \tag{1}$$

Being p the number that we want to check if it is a prime or not. And being m whatever integer number higher than $(p+1)/2$. If the result is zero (or it is so small that it can be considered to be zero) the number p is a prime. If not, the result will tell us how many permutations of two divisors, the number p has.

To obtain the integral (1), the following steps shall be taken. First, we will create a domain with all the composite numbers in the domain included.

Then, we will use the Fourier transform [1] [2] to change from the original domain (time domain) to the frequency domain.

Using the Parseval's theorem, we can check if a certain number is the above created domain or not. The use of Parseval's theorem will lead to the above integration.

If the number p that we want to check is not in the domain, the result of the integration is zero and the number is a prime. If instead, the result is an integer, this integer will tell us how many permutations of two divisors the number p has. And in consequence, how many divisors, the number has.

As k and ω are independent variables, the sum can be taken outside the integral. You can see that implementation in below Matlab/Octave program (**Figure 1**).

```
b=0;
p=input('Input a number : ');
m=fix((p+1)/2);
for k=2:m+1;
    fun=@(w)
exp(p.*1j.*w).*exp(-2.*k.*1j.*w).*(1-exp(-m.*k.*1j.*w))./(1-
exp(-k.*1j.*w));
    a=integral(fun,-pi,pi);
    b=b+a;
end;
b=b/(2*pi);
disp(b);
```

Figure 1. Matlab program to check if a number is prime using integration.

2. Matrix of Composite Numbers

It is very easy to create a matrix with the only composite numbers in certain domain. We can make that each element of the matrix is the result of the product of two integers in certain domain. We can perform this operation one by one until having used all the integers. So, the result will be all the composite numbers in certain domain.

You can see an example in **Figure 2**.

And the result attached in **Figure 3**.

We can see that all the composite numbers between 4 and 10 are included in the matrix: 4, 6, 8, 9, 10. And as expected, the prime numbers 2, 3, 5 and 7 not. But we can see that the number 14 that is composite is not included. Why? Because it is composed by 7×2 and the 7 is not included in the multiplications. This means, the domain where it is valid that all the composite numbers are included (and none of the integers) finishes in $2 \times 5 = 10$ (being 5 the last number we are using in the multiplications). We can also see that it is a symmetric matrix that will have implications in the following chapters.

We can increase the matrix to 7 for example and see the same effect (**Figure 4** and **Figure 5**).

As $2 \times 7 = 14$, we know that the result will be reliable until 14. This means, all the composite numbers until 14 have to be included in the matrix: 4, 6, 8, 9, 10, 12 and 14. And not the primes: 2, 3, 5, 7, 11, 13. We can see again that the composite number $22 = 11 \times 2$ is not included in the matrix. This is because 11 has not be included in the multiplications. The reliable domain is until $7 \times 2 = 14$, as commented.

The nomenclature that we will use in the following chapters for the general case will be the following (see **Figure 6**).

This means, the reliable domain (the domain where all the composite numbers will be included in the matrix) will be up to $2 \times (m+1)$.

$$\begin{bmatrix} 2\cdot 2 & 2\cdot 3 & 2\cdot 4 & 2\cdot 5 \\ 3\cdot 2 & 3\cdot 3 & 3\cdot 4 & 3\cdot 5 \\ 4\cdot 2 & 4\cdot 3 & 4\cdot 4 & 4\cdot 5 \\ 5\cdot 2 & 5\cdot 3 & 5\cdot 4 & 5\cdot 5 \end{bmatrix}$$

Figure 2. Matrix of the product of two integers within a certain domain.

$$\begin{bmatrix} 4 & 6 & 8 & 10 \\ 6 & 9 & 12 & 15 \\ 8 & 12 & 16 & 20 \\ 10 & 15 & 20 & 25 \end{bmatrix}$$

Figure 3. Matrix of the result of the product of two integers within a certain domain.

$$\begin{bmatrix} 2\cdot 2 & 2\cdot 3 & 2\cdot 4 & 2\cdot 5 & 2\cdot 6 & 2\cdot 7 \\ 3\cdot 2 & 3\cdot 3 & 3\cdot 4 & 3\cdot 5 & 3\cdot 6 & 3\cdot 7 \\ 4\cdot 2 & 4\cdot 3 & 4\cdot 4 & 4\cdot 5 & 4\cdot 6 & 4\cdot 7 \\ 5\cdot 2 & 5\cdot 3 & 5\cdot 4 & 5\cdot 5 & 5\cdot 6 & 5\cdot 7 \\ 6\cdot 2 & 6\cdot 3 & 6\cdot 4 & 6\cdot 5 & 6\cdot 6 & 6\cdot 7 \\ 7\cdot 2 & 7\cdot 3 & 7\cdot 4 & 7\cdot 5 & 7\cdot 6 & 7\cdot 7 \end{bmatrix}$$

Figure 4. Matrix of the products of 2 integers in the 2 to 7 domain.

$$\begin{bmatrix} 4 & 6 & 8 & 10 & 12 & 14 \\ 6 & 9 & 12 & 15 & 18 & 21 \\ 8 & 12 & 16 & 20 & 24 & 28 \\ 10 & 15 & 20 & 25 & 30 & 35 \\ 12 & 18 & 24 & 30 & 36 & 42 \\ 14 & 21 & 28 & 35 & 42 & 49 \end{bmatrix}$$

Figure 5. Matrix of the results of the products of two 2 integers in the 2 to 7 domain.

$$\begin{bmatrix} 2\cdot 2 & 2\cdot 3 & 2\cdot 4 & 2\cdot 5 & \cdots & 2\cdot(m+1) \\ 3\cdot 2 & 3\cdot 3 & 3\cdot 4 & 3\cdot 5 & \cdots & 3\cdot(m+1) \\ 4\cdot 2 & 4\cdot 3 & 4\cdot 4 & 4\cdot 5 & \cdots & 4\cdot(m+1) \\ 5\cdot 2 & 5\cdot 3 & 5\cdot 4 & 5\cdot 5 & \cdots & 5\cdot(m+1) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ (m+1)\cdot 2 & (m+1)\cdot 3 & (m+1)\cdot 4 & (m+1)\cdot 5 & \cdots & (m+1)\cdot(m+1) \end{bmatrix}$$

Figure 6. Matrix of the products of two integers in the 2 to $(m+1)$ domain.

3. Including the Elements of the Composite Matrix in the Discrete Time Domain

Now, the next movements will be quite straight forward. The first thing we have to do, is to introduce these calculated composite numbers in the discrete time domain. We will put unitary delta functions in the positions of the x axis indicated by the matrix of composite numbers (**Figure 7**).

We can see that in the above graphic, each Dirac delta represent the product of two integers, that compose a composite number. This means, each Dirac delta means that the composite number has a factorization of two numbers. And the order counts. This is, 2×4 counts one and 4×2 counts another one.

If we have only one Dirac Delta, it means that it has only one multiplication (therefore it is a square of a prime number) like $4 = 2 \times 2$ or $9 = 3 \times 3$ for example. If the number of Dirac deltas is odd, the number has an integer square root. If the number is even, it has different permutations of different products, but it does not have an integer square root.

We can create a function that is the sum of all these deltas. And this function will have the information of all these composite numbers. This function in the discrete domain is defined like this:

$$\begin{aligned} f(n) &= \sum_{k_1=2}^{k_1=m+1} \sum_{k_2=2}^{k_2=m+1} \delta\left(n-k_1\cdot k_2\right) \\ &= \sum_{k=2}^{k=m+1} \left(\delta(n-2\cdot k)+\delta(n-3\cdot k)+\delta(n-4\cdot k)+\cdots+\delta\left(n-(m+1)\cdot k\right)\right) \\ &= \delta(n-2\cdot 2)+\delta(n-2\cdot 3)+\cdots+\delta\left(n-2\cdot(m+1)\right) \\ &\quad +\delta(n-3\cdot 2)+\delta(n-3\cdot 3)+\cdots+\delta\left(n-(m+1)\cdot(m+1)\right) \end{aligned} \quad (2)$$

This function $f(n)$ has the information regarding all the composite numbers inside the domain $2 \times (m+1)$. But, the important information there is what it is not included. All the integers smaller than $2 \times (m+1)$ that are not included in this function, are prime numbers.

How do we spot them? If we want to know if the number 5 is prime or not. We have to check if there is a Dirac delta in position 5. This means, we have to

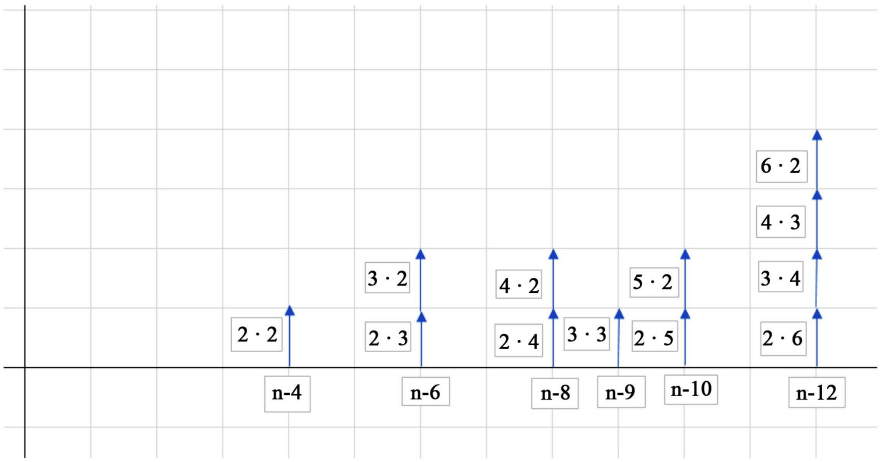

Figure 7. Representation in the time domain of the products of two integers, using Dirac delta functions.

check if the following function $g(n)$ is inside the function $f(n)$:

$$g(n)=\delta(n-5) \tag{3}$$

In general, if we want to check if whatever number p is prime, $g(n)$ would be:

$$g(n)=\delta(n-p) \tag{4}$$

We could do this performing the multiplication one by one for all the points in x axis between the function $f(n)$ and the function $g(n)$.

As the function $g(n)$ is zero everywhere except in the point 5 (or in general, in p), if the result of the multiplication point by point between $f(n)$ and $g(n)$ is zero, it means that $f(n)$ does not have a Dirac delta in 5. So, number 5 is prime.

If the result is different, it will tell us the value of $f(n)$ at that point. And the value of $f(n)$ at that point is the number of deltas included. This means, the number of all the possible multiplications of two integers that compose that number.

The problem is that to perform this operation in the time domain, does not add any value. Because to know the result, we have to compare "manually" $f(n)$ and $g(n)$ to see if $f(n)$ has a value different from zero in the position where $g(n)$ has the delta.

This is the reason why we go to the next chapter.

4. The Frequency Domain

What we will do is to convert the functions $f(n)$ an $g(n)$ to the frequency domain. To do so, we use the Fourier transform.

We start with $f(n)$. We know from signal theory that the Fourier transform of a shifted Dirac delta is:

$$\delta(n-a)\rightarrow \mathrm{e}^{-aj\omega} \tag{5}$$

So, the function $f(n)$, in the frequency domain has the form F:

$$\begin{aligned} F(\omega) &= \sum_{k_1=2}^{k_1=m+1} \sum_{k_2=2}^{k_2=m+1} \mathrm{e}^{-k_1 \cdot k_2 j\omega} \\ &= \mathrm{e}^{-2\cdot 2 j\omega} + \mathrm{e}^{-2\cdot 3 j\omega} + \cdots + \mathrm{e}^{-2\cdot (m+1) j\omega} \\ &\quad + \mathrm{e}^{-3\cdot 2 j\omega} + \mathrm{e}^{-3\cdot 3 j\omega} + \cdots + \mathrm{e}^{-3\cdot (m+1) j\omega} + \cdots \\ &\quad + \mathrm{e}^{-(m+1)\cdot 2 j\omega} + \mathrm{e}^{-(m+1)\cdot 3 j\omega} + \cdots + \mathrm{e}^{-(m+1)\cdot (m+1) j\omega} \end{aligned} \tag{6}$$

We can put above sum in the following form:

$$\begin{aligned} &= \left(\mathrm{e}^{-2 j\omega}\right)^2 + \left(\mathrm{e}^{-2 j\omega}\right)^3 + \cdots + \left(\mathrm{e}^{-2 j\omega}\right)^{m+1} \\ &\quad + \left(\mathrm{e}^{-3 j\omega}\right)^2 + \left(\mathrm{e}^{-3 j\omega}\right)^3 + \cdots + \left(\mathrm{e}^{-3 j\omega}\right)^{m+1} + \cdots \\ &\quad + \left(\mathrm{e}^{-(m+1) j\omega}\right)^2 + \left(\mathrm{e}^{-(m+1) j\omega}\right)^3 + \cdots + \left(\mathrm{e}^{-(m+1) j\omega}\right)^{m+1} \end{aligned}$$

As you can see above, each line is a geometric series sum. So, using the known result of geometric series [3] [4], we can write:

$$\begin{aligned} &= \mathrm{e}^{-2 j\omega} \frac{1-\mathrm{e}^{-2mj\omega}}{1-e^{-2 j\omega}} + \mathrm{e}^{-3 j\omega} \frac{1-\mathrm{e}^{-3mj\omega}}{1-\mathrm{e}^{-2 j\omega}} + \cdots + \mathrm{e}^{-(m+1) j\omega} \frac{1-\mathrm{e}^{-(m+1)mj\omega}}{1-\mathrm{e}^{-(m+1) j\omega}} \\ &= \sum_{k=2}^{k=m+1} \mathrm{e}^{-2kj\omega} \frac{1-\mathrm{e}^{-mkj\omega}}{1-\mathrm{e}^{-kj\omega}} \end{aligned} \tag{7}$$

For the function *g*, following the same process:

$$g(n) = \delta(n-p) \tag{4}$$

The Fourier transform would be:

$$g(n) = \delta(n-p) \to G(\omega) = \mathrm{e}^{-pj\omega} \tag{8}$$

5. Parseval's Theorem

Now, we get to the point. In the chapter 2, we wanted to perform the following operation:

$$\sum_{n=-\infty}^{n=\infty} f(n) \cdot g(n) = \sum_{n=4}^{n=(m+1)(m+1)} f(n) \cdot g(n) \tag{9}$$

If the result is different from zero, it means that *p* is composite, as it means that *g*(*n*):

$$g(n) = \delta(n-p) \tag{4}$$

Has found another delta in the same position for the function *f*(*n*) defined before, that has only deltas in composite number positions.

This operation in the time domain does not have added value (as you have to perform all multiplications), but fortunately Marc-Antoine Parseval [5] [6] found the following theorem.

The general Parseval's theorem applied to discrete functions (as it is the case) says the following:

$$\sum_{n=-\infty}^{n=\infty} f(n) \cdot g^*(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} F(\omega) \cdot G^*(\omega) \mathrm{d}\omega \tag{10}$$

The asterisk* means the complex conjugate. In all the functions we are dealing

in this paper are real in the time domain so in the left-hand side no conjugation is necessary.

$$g^*(n) = g(n) \tag{11}$$

For the right-hand side, yes, we have complex functions. But, as you can check all of them are exponential based. So, the conjugate is just to change the sign of the exponent. This is immediate from Euler's formula [7] [8].

$$G(\omega) = \mathrm{e}^{-pj\omega} \tag{12}$$

$$G^*(\omega) = \mathrm{e}^{pj\omega} \tag{13}$$

So, coming back, this means, we can perform the operation of multiplying point by point the two functions in the time domain, just making the definite integral form $-\pi$ to π of the multiplication of the two functions (its Fourier Transform) in the frequency domain.

So, the integral in the frequency domain (that represents the multiplication point by point of the functions $f(n)$ and $g(n)$ in the time domain) according Parseval's theorem is the following (and we will call q to its result):

$$\begin{aligned} q &= \sum_{n=-\infty}^{n=\infty} f(n)\cdot g^*(n) = \sum_{n=-\infty}^{n=\infty} f(n)\cdot g(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} F(\omega)\cdot G^*(\omega)\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum\nolimits_{k=2}^{k=m+1} \mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{-mkj\omega}}{1-\mathrm{e}^{-kj\omega}}\right)\cdot \mathrm{e}^{pj\omega}\mathrm{d}\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\left(\sum\nolimits_{k=2}^{k=m+1} \mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{-mkj\omega}}{1-\mathrm{e}^{-kj\omega}}\right)\mathrm{d}\omega \end{aligned} \tag{14}$$

So, summing up:

$$q = \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\left(\sum\nolimits_{k=2}^{k=m+1} \mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{-mkj\omega}}{1-\mathrm{e}^{-kj\omega}}\right)\mathrm{d}\omega \tag{14}$$

So, this is the integral we have been talking about from the beginning. Now, you know where it comes from. We will call q the result of above formula. If q is zero, p is prime, if q is different from zero, q has information regarding the number of factors. Let's comment about it in the following chapters.

Before that, just for possible calculation issues, it is to be noted that as, k is independent from ω the sum can be taken outside the integral, as follows:

$$\begin{aligned} q &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\left(\sum\nolimits_{k=2}^{k=m+1} \mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{-mkj\omega}}{1-\mathrm{e}^{-kj\omega}}\right)\mathrm{d}\omega \\ &= \frac{1}{2\pi}\sum_{k=2}^{k=m+1}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{-mkj\omega}}{1-\mathrm{e}^{-kj\omega}}\mathrm{d}\omega \end{aligned} \tag{15}$$

6. Number of Factors of *p*

We know that if p is composite, the number q (result from Equation (15)) is the number of possible combinations of multiplications of two integer numbers that get the value p. And yes, with this number we can get the number of factors that p has.

This is the formula:

$$q = (n_{f1}+1)(n_{f2}+1)(n_{f3}+1)\cdots(n_{fz}+1) - 2 \tag{16}$$

Being n_{f1} the number of times the factor $f1$ is repeated, n_{f2} the number of times $f2$ is repeated and so on.

Putting in another way is:

$$q + 2 = (n_{f1}+1)(n_{f2}+1)(n_{f3}+1)\cdots(n_{fz}+1) \tag{17}$$

So, the factors of the obtained number $q + 2$ (much smaller than p) tell us how many factors the number p originally has and how many of them are repeated.

Let's see with an example. We apply Equation (15) to the number 20 and we get $q = 4$. So we can check:

$$\begin{aligned} q + 2 = 4 + 2 = 6 &= (n_{f1}+1)(n_{f2}+1)(n_{f3}+1)\cdots(n_{fz}+1) \\ &= 3\times 2 = (2+1)(1+1) \end{aligned} \tag{18}$$

This means, p has to have a factor that is repeated twice and one that appears only once. And we see that it is true:

$$20 = 2\times 2\times 5 \tag{19}$$

The factor 2 is repeated twice and 5 once.

And the q is 4 because, there are four combinations of multiplications of two integers that get the result 20:

$$2\times 10, 4\times 5, 5\times 4, 10\times 2$$

So, everything fits.

Let's check, other ones:

$p = 102$

$q = 6$

$$q + 2 = 8 = 2\times 2\times 2 = (1+1)(1+1)(1+1) \tag{20}$$

This means, three factors that are not repeated, and in fact $102 = 2 \times 3 \times 17$.

Another one:

$p = 36$

$q = 7$

$$q + 2 = 9 = 3\times 3 = (2+1)(2+1) \tag{21}$$

This means two factors repeated twice (each). In fact, $36 = 2 \times 2 \times 3 \times 3$.

You can see that the Formula (17) works.

We can use q to know how many factors a number p has if it is composite (or instead, detect it is a prime if $q = 0$).

One of the problems that this system has is that sometimes there are different solutions for the number of factors. For example, for $q = 2$, the typical answer is:

$$q + 2 = 4 = 2\times 2 = (1+1)(1+1) \tag{22}$$

This means, two different factors not repeated. For example, $21 = 3 \times 7 = 7 \times 3$.

But it could be:

$$q+2=4=(3+1) \tag{23}$$

For example, the number $8 = 2 \times 4 = 4 \times 2$.

It has $q = 2$ (two combinations of factors) but $8 = 2 \times 2 \times 2$. This means one factor three times, as seen before.

So, the number q could have some values that have different solutions regarding the number of factors.

7. Efficiency of the Numerical Integration Compared to Brute Force Checking

We will check that the sum and the integral of the Formula (14) are very inefficient computation-wise. And in fact, it is more efficient to start looking for factors (by brute-force) than applying the formula. You can see this easily with the Matlab program (Figure 8).

```
b=0;
p=input('Input a number : ');
%using numerical integration
t1=cputime;
m=fix((p+1)/2);
for k=2:m+1;
    fun=@(w)
exp(p.*1j.*w).*exp(-2.*k.*1j.*w).*(1-exp(-m.*k.*1j.*w))./(1-exp(-k
.*1j.*w));
    a=integral(fun,-pi,pi,'AbsTol',0.24);
    b=b+a;
end;
b=b/(2*pi);
disp('------------------------------');
disp('Using numerical integration');
disp (num2str(b));
c=round(abs(b));
if c==0;

   disp([num2str(p),' is a prime number']);
else
   disp([num2str(p),' is a composite number and has ',num2strI,'
permutations of 2 factors']);
end

t1=cputime-t1;
disp([num2str(t1),' seconds']);

%Brute force checking
t2=cputime;
disp('------------------------------');
disp(['Using brute-force checking']);
if rem(p,2)==0;
   disp([num2str(p),' is a composite number and 2 is one of the
factors']);
else;
    k=3;
    c=sqrt(p);
    while k<=c;
      if mod (p,k)==0;
       disp([num2str(p),' is a composite number and ',num2str(k),' is
one of the factors']);
       k=c+10;
      else
         k=k+2;
      end
    end
    if k~=(c+10)
       disp([num2str(p),' is a prime number']);
    end
end
t2=cputime-t2;
disp([num2str(t2),' seconds'])
disp('------------------------------');
disp(['Numerical integration is ',num2str(t1/t2), ' times slower']);
```

Figure 8. Matlab program to compare integration method versus brute force checking.

If we check with different numbers, we compare the efficiency of each method (as time taken is recorded).

For the following examples, I will use a Windows 10 OMEN by HP Laptop 17-an0xx Intel-Core i7-7700HQ 2.80 GHz 32 Gb RAM Nvidia GeForce GTX 1070 with the program Matlab R2018b.

For example, with $p = 20$:

```
Input a number : 20
Using numerical integration
4-2.7134e-15i
20 is a composite number and has 4 permutations of 2 factors
0.03125 seconds
Using brute-force checking
20 is a composite number and 2 is one of the factors
0 seconds
Numerical integration is Inf times slower
```

With $p = 103$:

```
Input a number : 103
Using numerical integration
0.062599-2.3824e-13i
103 is a prime number
1.2031 seconds
Using brute-force checking
103 is a prime number
0 seconds
Numerical integration is Inf times slower
```

For $p = 300$:

```
Input a number : 300
Using numerical integration
16.1276+8.75419e-12i
300 is a composite number and has 16 permutations of 2 factors
12.2969 seconds
Using brute-force checking
300 is a composite number and 2 is one of the factors
0 seconds
Numerical integration is Inf times slower
```

We can see that for $p = 300$, it has taken 12.25 seconds to discover that has 16 permutations of two factors. Which leads to $q+2=18=(2+1)(2+1)(1+1)$.

This means that has three factors (and two of them are repeated twice). (In fact, these are = 2 × 2 × 3 × 5 × 5).

But with brute-checking it took zero seconds to discover it was composite and even it has given the info of which is one of the factors (the factor 2).

Regarding this point a very detailed study should be done. You can check in the Matlab program that I have reduced the precision of the integral to increase efficiency. I have reduced the precision of 0.24, which can be considered high, but it is sufficient for low values of p.

You can change and play with the value here:

a = integral(fun,-pi,pi,'AbsTol',0.24);

Also, I have considered implicitly that q is considered to be zero when $q < 0.5$ (using the function "round").

8. Sum of the Exponential of the Factors of the Number *p*

In the chapter 2 we have put all the Dirac deltas as unitary Dirac deltas. But we can introduce more information, if instead of putting unitary Dirac deltas, we introduce the deltas multiplied by a number that represents the value of the factors you are using. This means, as an output we can get info regarding the factors of the number p (not only the quantity of factors but its sum).

We can check an example (see Figure 9). We can multiply the delta by an exponential that represents the two factors we are using. In this case, it is the exponential of the sum of the two factors. We will see that this representation gives us more information and permits to follow the calculations in a similar manner as before.

So, in this case:

$$\begin{aligned} f(n) &= \sum_{k_1=2}^{k_1=m+1}\sum_{k_2=2}^{k_2=m+1} \mathrm{e}^{k_1}\mathrm{e}^{k_2}\delta\left(n-k_1\cdot k_2\right) \\ &= \sum_{k=2}^{k=m+1}\left(\mathrm{e}^2\mathrm{e}^k\delta\left(n-2\cdot k\right)+\mathrm{e}^3\mathrm{e}^k\delta\left(n-3\cdot k\right)+\mathrm{e}^4\mathrm{e}^k\delta\left(n-4\cdot k\right)+\cdots\right. \\ &\quad \left.+\mathrm{e}^{n-(m+1)}\mathrm{e}^k\delta\left(n-(m+1)\cdot k\right)\right) \\ &= \mathrm{e}^{2+2}\delta\left(n-2\cdot 2\right)+\mathrm{e}^{2+3}\delta\left(n-2\cdot 3\right)+\cdots+\mathrm{e}^{2+(m-1)}\delta\left(n-2\cdot(m+1)\right) \\ &\quad +\mathrm{e}^{3+2}\delta\left(n-3\cdot 2\right)+\mathrm{e}^{3+3}\delta\left(n-3\cdot 3\right)+\cdots+\mathrm{e}^{m+1+m+1}\delta\left(n-(m+1)\cdot(m+1)\right) \end{aligned} \tag{25}$$

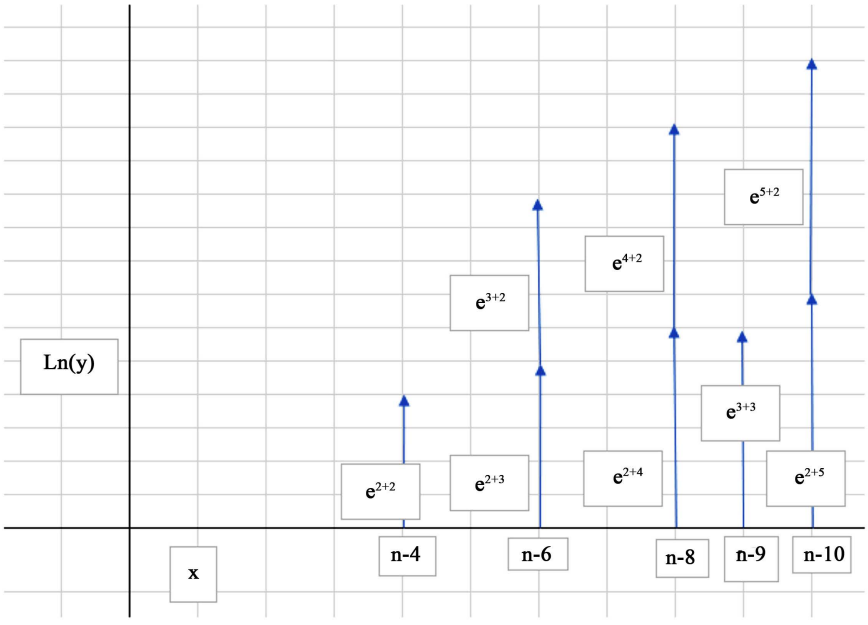

Figure 9. Representation in the time domain of the products of the exponential of two integers, using delta functions.

To calculate Fourier transform:

$$\begin{aligned}F(\omega) &= \sum_{k_1=2}^{k_1=m+1}\sum_{k_2=2}^{k_2=m+1} \mathrm{e}^{k_1}\mathrm{e}^{k_2}\mathrm{e}^{-k_1\cdot k_2 j\omega} \\ &= \mathrm{e}^{2+2}\mathrm{e}^{-2\cdot 2 j\omega} + \mathrm{e}^{2+3}\mathrm{e}^{-2\cdot 3 j\omega} + \cdots + \mathrm{e}^{2+(m-1)}\mathrm{e}^{-2\cdot(m+1)j\omega} \\ &\quad + \mathrm{e}^{3+2}\mathrm{e}^{-3\cdot 2 j\omega} + \mathrm{e}^{3+3}\mathrm{e}^{-3\cdot 3 j\omega} + \cdots + \mathrm{e}^{3+m+1}\mathrm{e}^{-3\cdot(m+1)j\omega} + \cdots \\ &\quad + \mathrm{e}^{m+1+2}\mathrm{e}^{-(m+1)\cdot 2 j\omega} + \mathrm{e}^{m+1+3}\mathrm{e}^{-(m+1)\cdot 3 j\omega} + \cdots + \mathrm{e}^{m+1+m+1}\mathrm{e}^{-(m+1)\cdot(m+1)j\omega}\end{aligned}$$

Calculating the geometric series result for each line:

$$\begin{aligned}&\mathrm{e}^2\mathrm{e}^{-2j\omega}\frac{1-\mathrm{e}^{m-2mj\omega}}{1-\mathrm{e}^{1-2j\omega}} + \mathrm{e}^3\mathrm{e}^{-3j\omega}\frac{1-\mathrm{e}^{m-3mj\omega}}{1-\mathrm{e}^{1-2j\omega}} + \cdots + \mathrm{e}^{m+1}\mathrm{e}^{-(m+1)j\omega}\frac{1-\mathrm{e}^{m-(m+1)mj\omega}}{1-\mathrm{e}^{1-(m+1)j\omega}} \\ &= \sum_{k=2}^{k=m+1}\mathrm{e}^k\mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{m-mkj\omega}}{1-\mathrm{e}^{1-kj\omega}} = \sum_{k=2}^{k=m+1}\mathrm{e}^{k(1-2j\omega)}\frac{1-\mathrm{e}^{m-mkj\omega}}{1-\mathrm{e}^{1-kj\omega}}\end{aligned} \tag{26}$$

$$\begin{aligned}q_2 &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\left(\sum_{k=2}^{k=m+1}\mathrm{e}^{k(1-2j\omega)}\frac{1-\mathrm{e}^{m-mkj\omega}}{1-\mathrm{e}^{1-kj\omega}}\right)\mathrm{d}\omega \\ &= \frac{1}{2\pi}\sum_{k=2}^{k=m+1}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\mathrm{e}^{k(1-2j\omega)}\frac{1-\mathrm{e}^{m-mkj\omega}}{1-\mathrm{e}^{1-kj\omega}}\mathrm{d}\omega\end{aligned} \tag{27}$$

We call q_2 this value that has this information regarding the factors. We will see how to use it in the next chapter.

One of the problems that this form has is that the exponentials ek and em could go high very quick. A way to avoid this, is to use a "calibration factor" *c*. This factor, will reduce this exponentials value not reducing the information. This factor should be of the order of *p*, and normally using the value $\sqrt{p}$ works well. This we will see later. First, the calculation (similar to above but with the calibration factor c included).

$$\begin{aligned}F(\omega) &= \sum_{k_1=2}^{k_1=m+1}\sum_{k_2=2}^{k_2=m+1} \mathrm{e}^{\frac{k_1}{c}}\mathrm{e}^{\frac{k_2}{c}}\mathrm{e}^{-k_1\cdot k_2 j\omega} \\ &= \mathrm{e}^{\frac{2+2}{c}}\mathrm{e}^{-2\cdot 2 j\omega} + \mathrm{e}^{\frac{2+3}{c}}\mathrm{e}^{-2\cdot 3 j\omega} + \cdots + \mathrm{e}^{\frac{2+(m-1)}{c}}\mathrm{e}^{-2\cdot(m+1)j\omega} \\ &\quad + \mathrm{e}^{\frac{3+2}{c}}\mathrm{e}^{-3\cdot 2 j\omega} + \mathrm{e}^{\frac{3+3}{c}}\mathrm{e}^{-3\cdot 3 j\omega} + \cdots + \mathrm{e}^{\frac{3+m+1}{c}}\mathrm{e}^{-3\cdot(m+1)j\omega} + \cdots \\ &\quad + \mathrm{e}^{\frac{m+1+2}{c}}\mathrm{e}^{-(m+1)\cdot 2 j\omega} + \mathrm{e}^{\frac{m+1+3}{c}}\mathrm{e}^{-(m+1)\cdot 3 j\omega} + \cdots + \mathrm{e}^{\frac{m+1+m+1}{c}}\mathrm{e}^{-(m+1)\cdot(m+1)j\omega}\end{aligned}$$

Calculating the geometric series result for each line:

$$\begin{aligned}&\mathrm{e}^{\frac{2}{c}}\mathrm{e}^{-2j\omega}\frac{1-\mathrm{e}^{\frac{m}{c}-2mj\omega}}{1-\mathrm{e}^{\frac{1}{c}-2j\omega}} + \mathrm{e}^{\frac{3}{c}}\mathrm{e}^{-3j\omega}\frac{1-\mathrm{e}^{\frac{m}{c}-3mj\omega}}{1-\mathrm{e}^{\frac{1}{c}-2j\omega}} + \cdots + \mathrm{e}^{\frac{m+1}{c}}\mathrm{e}^{-(m+1)j\omega}\frac{1-\mathrm{e}^{\frac{m}{c}-(m+1)mj\omega}}{1-\mathrm{e}^{\frac{1}{c}-(m+1)j\omega}} \\ &= \sum_{k=2}^{k=m+1}\mathrm{e}^{\frac{k}{c}}\mathrm{e}^{-2kj\omega}\frac{1-\mathrm{e}^{\frac{m}{c}-mkj\omega}}{1-\mathrm{e}^{\frac{1}{c}-kj\omega}} = \sum_{k=2}^{k=m+1}\mathrm{e}^{k\left(\frac{1}{c}-2j\omega\right)}\frac{1-\mathrm{e}^{\frac{m}{c}-mkj\omega}}{1-\mathrm{e}^{\frac{1}{c}-kj\omega}}\end{aligned} \tag{28}$$

$$\begin{aligned}q_3 &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\left(\sum_{k=2}^{k=m+1}\mathrm{e}^{\frac{k}{c}(1-2j\omega)}\frac{1-\mathrm{e}^{\frac{m}{c}-mkj\omega}}{1-\mathrm{e}^{\frac{1}{c}-kj\omega}}\right)\mathrm{d}\omega \\ &= \frac{1}{2\pi}\sum_{k=2}^{k=m+1}\int_{-\pi}^{\pi}\mathrm{e}^{pj\omega}\mathrm{e}^{\frac{k}{c}(1-2j\omega)}\frac{1-\mathrm{e}^{\frac{m}{c}-mkj\omega}}{1-\mathrm{e}^{\frac{1}{c}-kj\omega}}\mathrm{d}\omega\end{aligned} \tag{29}$$

This factor q_3 is equivalent (has the same information) as q_2 but reducing the exponentials value, so it reduces the computing necessities.

We see what to do with them, in the next chapter.

9. Special Case: Discovering the Factors of Two-Factor Composite Numbers

There are two cases where with all this information, it is immediate to calculate the factors of a number. One of the cases is trivial. When $q = 1$, it means that p is a perfect square, so you can just make the square root to get it.

But there is another special case. When $q = 2$, it means that the number is composite but only by two factors (two prime factors). This case is used for example in RSA encryption [9].

What you can do in this case is the following.

- First you calculate the q. And you see that is 2, that corresponds to (1 + 1)(1 + 1). This means, two different factors. So, you can apply the following.
- Then you calculate the q_2 according (27). Or calculate q_3 according (29).
- Calculate the following number using q_2:

$$d = Ln\left(\frac{q_2}{2}\right) + 2 \tag{30}$$

Or if you have calculated q_3:

$$d = Ln\left(\frac{q_2}{2}\right) \cdot c + 2 \tag{31}$$

Being c the calibration factor you have chosen.

- And, then solve the following equation system (being a and b the unknown factors of p) and d is the known number calculated before:

$$a \cdot b = p \tag{32}$$

$$a + b = d \tag{33}$$

You have the solution:

$$b = \frac{d + \sqrt{d^2 - 4p}}{2} \tag{34}$$

$$a = \frac{p}{b} = d - b = \frac{d - \sqrt{d^2 - 4p}}{2} \tag{35}$$

So, you can get, immediately the factors a and b. This, of course is tricky. As for the calculation of q_2 you have used exponentials, summations and numerical integrals. So, the complete process, as commented, is at this stage is inefficient.

If you do not believe all this, put this program in your Matlab/Octave (**Figure 10** and **Figure 11**). As commented, we have used the calibration factor c as $\sqrt{p}$.

You can see the following examples result:

```
b=0;
p=input('Input a number : ');
m=fix((p+1)/2);
for k=2:m+1;
    fun=@(w)
exp(-p.*1j.*w).*exp(2.*k.*1j.*w).*(1-exp(m.*k.*1j.*w))./(1-
exp(k.*1j.*w));
    a=integral(fun,-pi,pi);
    b=b+a;
end;
b=b/(2*pi);
if round(abs(b))==2;
b=0;
t1=cputime;
c=round(sqrt(p));
m=fix((p+1)/2);
for k=2:m+1;
    fun=@(w)
exp(p.*1j.*w).*exp((k./c-2.*k.*1j.*w)).*(1-exp(m./c-m.*k.*1
j.*w))./(1-exp(1./c-k.*1j.*w));
    a=integral(fun,-pi,pi);
    b=b+a;
end;
b=b/(2*pi);
disp('---------------------------');
disp(['Using numerical integration for p= ',num2str(p)]);
disp(['q2= ',num2str(b)]);
d=round((log(round(abs(b))/2)).*c+2);
disp(['Sum of factors ',num2str(d)]);
b=(d+sqrt(d^2-4*p))/2;
a=(d-b);
t1=cputime-t1;
disp(['Factors = ',num2str(a),' and ',num2str(b)]);
disp([num2str(t1),' seconds'])
```

Figure 10. Matlab program to obtain two factors using integration.

```
%Brute force checking
t2=cputime;
disp('------------------------------');
disp(['Using brute-force checking']);
if rem(p,2)==0;
    disp([num2str(p),' is a composite number and 2 is one
of the factors']);
else;
    k=3;
    c=sqrt(p);
    while k<=c;
      if mod (p,k)==0;
       disp([num2str(p),'  is  a  composite  number  and
',num2str(k),' is one of the factors']);
        k=c+10;
      else
          k=k+2;
      end
    end
    if k~=(c+10)
       disp([num2str(p),' is a prime number']);
    end
end
t2=cputime-t2;
disp([num2str(t2),' seconds'])
disp('------------------------------');
disp(['Numerical integration is',num2str(t1/t2), 'times
slower']);
else
    disp('this is not a two factor composite number');
end;
```

Figure 11. Matlab program to obtain two factors using brute-force checking.

Number $p = 21$:

```
----------------------------
Using numerical integration for p= 21
q2= 9.9061-6.2334e-14i
Sum of factors 10
Factors = 3 and 7
0.078125 seconds
------------------------------
Using brute-force checking
21 is a composite number and 3 is one of the factors
0.046875 seconds
------------------------------
Numerical integration is 1.6667 times slower
```

Number $p = 77$:

```
----------------------------
Using numerical integration for p= 77
q2= 11.8334+6.15351e-12i
Sum of factors 18
Factors = 7 and 11
10.2656 seconds
------------------------------
Using brute-force checking
77 is a composite number and 7 is one of the factors
0 seconds
------------------------------
Numerical integration is Inf times slower
```

Number 323 is especially painful:

```
----------------------------
Using numerical integration for p= 323
q2= 13.2246-1.5893e-06i
Sum of factors 36
Factors = 17 and 19
35 seconds
------------------------------
Using brute-force checking
323 is a composite number and 17 is one of the factors
0 seconds
------------------------------
Numerical integration is Inf times slower
```

We can see that still the brute-force checking is better computer efficiency-wise than using the numerical integration.

10. Future Developments

A deeper study shall be done regarding the efficiency of the summations and the integrals. Finding an analytical solution for one of both would be the key.

In the calculation of q_2, if use negative exponents for the real part, we could go to a summation that has zero in the infinity for the last element. This means we could simplify the Equation (but no improvement got at this stage).

It could be used the Euler-Maclaurin [10] equation to convert the integral to

another summation (but no improvement has been found at this stage compared to the numerical integration).

Also, the possibility of making some prework in Equation (1) as the summation is independent of p, should be checked. But regretfully the later integration, element by element, cannot be avoided anyhow at this stage.

11. Conclusions

In this paper, it has been shown a way of checking if a number is prime or not, using numerical integration in the complex plane. At this stage, this way of calculation is inefficient compared to brute-force checking. Only if in the future an analytical solution for the integral or some prework is done, this efficiency could increase.

Also, it has been used the same method to calculate the factors for the special case where a number is composed by only two factors. Regretfully, again, the procedure is more inefficient than a brute-force checking at this stage. But it opens a new field of calculation.

A deeper study shall be done regarding the efficiency of the summations and the integrals. Finding an analytical solution for one of both would be the key. Some improvements as trying to make the sum to infinity or converting it in an integral via Euler-Maclaurin equation have been discussed. Also, the probability of making some prework in the integral or in the sum before applying to the specific number to be checked shall be studied.

Bilbao, 19th January 2019 (JAMP-v2).

Acknowledgements

To my family and friends.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Fourier, J.B.J. (1878 [1822]) The Analytical Theory of Heat. Freeman, A., Trans., The University Press, Cambridge, United Kingdom. (Translated from French)

[2] https://en.wikipedia.org/wiki/Fourier_transform

[3] Euclid's Elements, Book IX, Proposition 35. Aleph0.clarku.edu.

[4] https://en.wikipedia.org/wiki/Geometric_series

[5] des Chênes, P. (1799) Marc-Antoine Mémoire sur les séries et sur l'intégration complète d'une équation aux différences partielles linéaire du second ordre, à coefficients constants. *Sciences, mathématiques et physiques* (*Savants étrangers*), **1**, 638-648.

[6] http://www.astro.rug.nl/~vdhulst/SignalProcessing/Hoorcolleges/college03.pdf

[7] Hazewinkel, M. (Ed.) (2001) Euler Formulas. Encyclopaedia of Mathematics, Springer.

[8] https://en.wikipedia.org/wiki/Euler%27s_formula

[9] Rivest, R., Shamir, A. and Adleman, L. (1978) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, **21**, 120-126. https://doi.org/10.1145/359340.359342

[10] Apostol, T.M. (1999) An Elementary View of Euler's Summation Formula. The American Mathematical Monthly. *Mathematical Association of America*, **106**, 409-418.