

Composite Mersenne Numbers -
A System for the Derivation
of Their Factors.

Peter G.Bass.

ABSTRACT.

This paper presents new methods for the determination of the factors of composite Mersenne numbers, which are then also provided in the form of five Python programs.

CONTENTS.

1.0 Introduction.

2.0 Nomenclature.

3.0 Analytical Preamble.

- 3.1 Basis of the Analysis.
- 3.2 Composites Derivation via Exponent Extrapolation when p is Composite.
- 3.3 Derivation of the Maximum Possible Value of the Smallest Factor.
- 3.4 The Iteration Value of Test Factors.
- 3.5 Determination of Permissible Values of n .

4.0 Python Programs.

- 4.1 Determination of All Exponents and Initial Factors of Mersenne Composites up to a maximum Value of the Exponent, (Program #1).
- 4.2 Determination of the Exponent Extrapolation Value for a Single Mersenne Composite, (Program #2).
- 4.3 Determination of the Initial Factor of a Single Mersenne Composite when the Exponent is Prime, (Program #3).
- 4.4 For a Single Factor, Determination of the Exponents of the Multiple Associated Mersenne Composites, (Program #4).
- 4.5 For a Single Exponent, Determination of All the Factors of the Associated Mersenne Composite, (Program #5).
- 4.6 Examples of Program Application.
 - 4.6.1 Program #1.
 - 4.6.2 Program #2.
 - 4.6.3 Program #3.
 - 4.6.4 Program #4.
 - 4.6.5 Program #5.

5.0 Conclusions.

APPENDICES.

- A.1 Analytical Factorisation of Mersenne Composites when the Exponent is Composite.
- A.2 Fractional Values of k in Factors of Mersenne Composites when $f = 2kp + 1$.
- A.3 Truth Tables for Program #2.

REFERENCES.

1.0 Introduction.

Mersenne numbers are named after the French monk Marin Mersenne, who studied them in the 17th Century. They were however, also well known to the ancient Greek mathematicians, circa 500 to 275 BC and others prior to Mersenne. Mersenne numbers take the form

$$M_p = 2^p - 1 \quad (1.1)$$

where p , the exponent, can take any positive integer value.

Modern investigations of Mersenne numbers are primarily aimed at finding record breaking prime values of M_p . It is well known that for such numbers, p must also be prime. These Mersenne primes contain many millions of digits and their determination is the objective of the GIMPS Project, (Great Internet Mersenne Prime Search). In this project, many ordinary personal computers are connected via the internet to a master server, which allocates various tasks to them. Candidate Mersenne numbers, as thereby identified, are then tested using special software to determine whether they are prime or composite.

It is the purpose of this paper to present simple means of determining the factors of composite Mersenne numbers. This task is to be viable on a single computer. However, some composite Mersenne numbers exhibit many factors. Also, each factor will be the factor of an infinite number of Mersenne composites, with a very large degree of overlap. Consequently, it is not possible to derive all of these composites for each factor. Instead here, a single factor is determined for each of a selected number of Mersenne composites, and the methods used will then show how to obtain, if desired, many others.

Note that in this paper, only odd values of p are considered and also, Mersenne primes are not specifically addressed.

2.0 Nomenclature.

M	A Mersenne Number.
p	The Exponent of a Mersenne Number.
N	A Natural Number
m	An Arbitrary Variable.
n	An Arbitrary Variable.
f	A Factor of a Mersenne Number.
i, j	A Mersenne Number Identifier.
q	A Mersenne Exponent Additive Variable.
u	An Exponent Identifier.
k	An Arbitrary Number in a Factor of a Mersenne Number.
$\pi(N)$	The Number of Primes from 0 to N .
a	A Factor of p .
b	A Factor of p .

3.0 Analytical Preamble.

3.1 Basis of the Analysis.

The primary relationship used here is taken from [1], Eq.(2.5), the salient part of which is repeated here for convenience,

$$N = (2m + 1)\{2(n - 1) + (2m + 1)\} \quad (3.1)$$

and where clearly the term $(2m + 1)$ is the first factor of N . In [1] (3.1) was used to determine all the odd composite numbers, and thereby all the primes, up to some maximum value of N . To use (3.1) to determine the composites of specific types of numbers, such as (1.1), it is rewritten as

$$M_p = 2^p - 1 = f\{2(n-1) + f\} \quad (3.2)$$

where f is a low prime factor. Rearranging (3.2) to

$$n = \frac{2^p - (f-1)^2}{2f} \quad (3.3)$$

Either p or f can now be sequentially varied, (with the other parameter constant), until n becomes an integer, which then confirms that f is a factor of M_p . A second, and possibly further factors, may then be obtained from the method in Section 3.2 below. Such f or p iterations are of course best performed on a computer, and are such that access to a large database of prime numbers is not necessary. Subsequent values of p for which f is also a low prime factor of M_p can then also be obtained from the method in Section 3.2 below. Note that because only odd values of p are considered here, the minimum value of f for any M_p is 7.

3.2 Composites Derivation Via Exponent Extrapolation when p is Composite.

When it is known that p is composite, further values of p of which f is a factor of the associated Mersenne number can be obtained by exponent extrapolation as follows.

Case (i). With

$$M_i = f_1 f_2 = 2^p - 1 \quad (3.4)$$

and

$$M_j = f_1 f_3 = 2^{p+q} - 1 \quad (3.5)$$

then

$$\frac{f_3}{f_2} = \frac{2^{p+q} - 1}{2^p - 1} \quad (3.6)$$

Carrying out the division

$$\begin{array}{r} 2^q + 2^{(q-p)} + \dots + 1 \\ \underline{2^p - 1) 2^{p+q} - 1} \\ 2^{p+q} - 2^q \\ - \quad \underline{2^q - 1} \\ 2^q - 2^{(q-p)} \\ - \quad \underline{2^{(q-p)} - 1} \\ \vdots \\ - \quad \underline{2^{(q-(m-1)p)} - 1} \\ 2^p - 1 \\ - \quad - \end{array} \quad (3.7)$$

The final term occurs when $q = mp$ with m an integer, so that

$$2^{(p+q)} = 2^{(m+1)p} \quad (3.8)$$

and therefore f_1 is a factor of $2^{(m+1)p} - 1$ where m is zero or, because only odd values of p are being considered, any even positive integer.

Case (ii). A special case occurs when f_1 is itself also a Mersenne number, thus

$$f_1 = 2^m - 1 \quad (3.9)$$

Then in (3.5)

$$f_3 = \frac{2^{p+q} - 1}{2^m - 1} \quad (3.10)$$

Carrying out the division

$$\begin{array}{r} 2^{p+q-m} \\ \hline 2^m - 1 \overline{) 2^{p+q} - 1} \\ \underline{2^{p+q} - 2^{p+q-m}} \\ - \quad 2^{p+q-m} - 1 \end{array} \quad (3.11)$$

and f_3 can only be an integer if $q = m$ so that (3.11) becomes

$$\begin{array}{r} 2^p + f_2 \\ \hline 2^m - 1 \overline{) 2^{p+q} - 1} \\ \underline{2^{p+q} - 2^p} \\ - \quad \underline{2^p - 1} \\ \quad \underline{2^p - 1} \\ \quad - \quad - \end{array} \quad (3.12)$$

so that

$$f_3 = 2^p + f_2 \quad (3.13)$$

Now if p_u is a large composite exponent, then from (3.10) and (3.13)

$$2^p + f_2 = \frac{2^{p_u} - 1}{2^m - 1} \quad (3.14)$$

Substituting for f_2

$$2^p + \frac{2^p - 1}{2^m - 1} = \frac{2^{p_u} - 1}{2^m - 1} \quad (3.15)$$

Which reduces to

$$2^{p+m} - 1 = 2^{p_u} - 1 \quad (3.16)$$

So that

$$p_u = m + p \quad (3.17)$$

and therefore, because only odd values of p are being considered, this becomes

$$p_u = p + 2m \quad (3.18)$$

Therefore if $f_1 = 2^m - 1$ is a factor of $2^p - 1$ then it is also a factor of $2^{(p+2m)} - 1$. Also note that accordingly, if $2^m - 1$ is composite, then its factors are also factors of $2^{p_u} - 1$. This comment applies to both cases of this Section.

Extrapolation values follow a simple law. If $f_1 = 2^m - 1$, then the extrapolation value is $2m$. For all other factors, the extrapolation value is $2p$ where p is the lowest exponent for which f_1 is a factor of the associated Mersenne composite.

3.3 Determination of the Maximum Possible Value of the Smallest Factor.

When determining the factors of any Natural number N , the largest possible factor is always the nearest integer to $N^{1/2}$. This is not the case with Mersenne numbers however, because of their form. The largest possible value of the smallest factor is determined as follows. If

$$M_p = 2^p - 1 \quad (3.19)$$

then

$$M_p = \left(2^{(p-1)/2} + 1\right)\left(2^{(p+1)/2} - 1\right) - 2^{(p+1)/2} + 2^{(p-1)/2} \quad (3.20)$$

$$= \left(2^{(p-1)/2} + 1\right)\left(2^{(p+1)/2} - 1\right) + 2^{(p-1)/2}(-2 + 1)$$

$$= \left(2^{(p-1)/2} + 1\right)\left(2^{(p+1)/2} - 1\right) - 2^{(p-1)/2}$$

$$= \left(2^{(p-1)/2} + 1\right)\left\{2^{(p+1)/2} - 1 - \frac{2^{(p-1)/2}}{2^{(p-1)/2} + 1}\right\} \quad (3.21)$$

The quotient term in the large brackets can be approximated to unity so that

$$M_p \approx \left(2^{(p-1)/2} + 1\right)\left(2^{(p+1)/2} - 2\right) = M_p - 1 \quad (3.22)$$

This is the closest that two integers can approximate the factors of M_p . Thus the largest possible value of the smallest factor is

$$\left(2^{(p-1)/2} + 1\right) \quad (3.23)$$

Note that this also means that $\left(2^{(p+1)/2} - 2\right)$ is the smallest possible value of the largest factor of M_p .

3.4 The Iteration Value of Test Factors.

In program #1, with f starting at 7 there are only two iteration values that can be used. They are 2 and 8. Any other values will result in false primes being identified. While 8 can be used as an iteration value it will not always produce the lowest factor. For instance, in determining the factor of $2^{29} - 1$ a factor of 1103 is produced, which while correct, is not the lowest which is 233. Therefore, the only iteration value which produces consistently accurate lowest prime results is 2.

For values of p that can be easily factored, the low factor of M_p can be determined by inspection. i.e. if p is divisible by a then M_p is divisible by $2^a - 1$. The following table gives a few examples.

p	Factor of p	Factor of M_p
9	3	$7 = 2^3 - 1$
25	5	$31 = 2^5 - 1$
49	7	$127 = 2^7 - 1$
221	13	$8191 = 2^{13} - 1$
323	17	$131071 = 2^{17} - 1$
1299	3	$7 = 2^3 - 1$

Table 3.1 – Factors of p to Factors of M_p .

However, note that these may not be the lowest factors.

Also it is well known that the factors of a composite Mersenne number can be given by

$$f = 2kp + 1 \quad (3.24)$$

This relationship is only used in this paper when p is prime because when p is composite, it does not always produce a satisfactory result, (see Appendix B).

3.5 Determination of Permissible Values of n .

Using (3.2)

$$f\{2(n-1) + f\} = 2^p - 1 \quad (3.25)$$

If $n = 0$, then

$$f^2 - 2f + 1 = 2^p \quad (3.26)$$

so that

$$f = \left(2^{p/2} \pm 1 \right) \quad (3.27)$$

and because $2^{p/2} \pm 1$ is not an integer, (p being odd), then $n = 0$ is not a permissible value.

If n is negative, then (3.26) becomes

$$f^2 - 2f(n+1) + 1 - 2^p = 0 \quad (3.28)$$

solving for f

$$f = n + 1 + \left(n^2 + 2n + 2^p \right)^{1/2} \quad (3.29)$$

For f to be an integer, the term in the brackets must be a perfect square. While this is possible, it merely provides the same value of f , for the same value of p , as obtained with a different, positive, value of n . Therefore, only positive values of n need be considered.

4.0 Python Programs.

The Python program was chosen for this exercise because it contains arbitrary precision integer arithmetic, a facility necessary for computation with very large numbers. However, to run the programs here, it is also necessary to install the floating point equivalent.

The codes below should be entered into Python's IDLE facility and run from there or from the saved *.py executable.

4.1 Determination of all Exponents and Initial Factors of Mersenne Composites Up to a Maximum Value of the Exponent, (Program #1).

This program embodies the results of Section 3.1, Eq.(3.3), Section (3.3), Eq. (3.23) and Section 3.4.

```
print("\n\nTo Find the First Factor from the Exponents.\n\n")

import time
import math
from mpmath import mp
from mpmath import mpf

pmax=int(input("\nEnter Maximum Value of Exponent, pmax:- "))
mp.dps=int(pmax/3)
p=int(input("\nEnter Start Value of Exponent. Must be less than or equal to pmax:- "))
print("\n\n")

f=7
fmax=8
count=0

start=time.clock()

while p<=pmax:
    fmax=int(2**((p-1)/2)+1)
    while f<fmax:
        n=mpf(2**p-(f-1)**2)/(2*f)
        if n==int(n):
            print("p is",p),
            print("f is",f)
            break
        else:
            f=f+2
            count +=1
            if count==1000000:
                print("p is", p),
                print("and 2^p-1, is Possibly Prime.")
                break
        f=7
        p=p+2
        count=0

finish=time.clock()
run=finish-start
hours=int(run/3600)
run=run-hours*3600
minutes=int(run/60)
seconds=run-minutes*60
```

```
print("\n\nRun Time is:- ")
print(hours, minutes, seconds)

input("\n\npress the enter key to exit. ")
```

Table 4.1 – Program #1, To Find the First Factor from the Exponents.

This program will print, (on screen), the initial factor and first exponent of each Mersenne composite up to the value of p_{max} specified. The minimum start value of p must be 9. To compute for a mid-range, enter a start value greater than 9, or to find the first factor of a single Mersenne number, enter the same value for p and p_{max} . Note that to minimise computation time, a termination control has been included should a value of p be encountered for which the Mersenne number is prime, or for which the low factor is an exceptionally large prime. In these cases, the value of p should be checked using program #3 in conjunction with the Lucas-Lehmer test method.

4.2 Determination of the Exponent Extrapolation Value for a Single Mersenne Composite, (Program #2).

This program embodies the results of Section 3.2. The process is only really useful if a factor of a particular Mersenne composite is already known, and it is desired to find the range of Mersenne composites to which this factor applies. It is very important to distinguish results applicable to the two cases discussed in Section 3.2.

```
print("\nThis Program Determines the Extrapolation Values, ")
print("for any Particular Mersenne Number Exponent.\n")

import time
import math
from mpmath import mp
from mpmath import mpf

p=int(input("\nEnter Value of Exponent:- "))
mp.dps=int(p/3)
m=int(input("\nEnter Value of m, Must be Odd and Minimum is 3:- "))

q=0

start=time.clock()

while m<=p:
    q=mpf(p)/m
    if q<=2:
        print("\nNo Result, p is Prime ")
        break
    elif q==int(q):
        print("\np is Composite.\n ")
        print("The p Extrapolation Value is "),
        print(2*m),
        break
    else:
        m=m+2
```

```

finish=time.clock()
run=finish-start
hours=int(run/3600)
run=run-hours*3600
minutes=int(run/60)
seconds=run-minutes*60

print("\n\nRun Time is:- ")
print(hours, minutes, seconds)

input("\n\npress the enter key to exit. ")

```

Table 4.2 – Program #2, Exponent Extrapolation Determination.

If p is composite, this program will print, (on screen), the value of the exponent extrapolation value.

If not already known, the initial factor of the associated Mersenne composite can be obtained from the output of the program in Section 4.1 (to enable the creation of the tables in Section 4.6.1).

Having found the first extrapolation value for some m_1 , if desired the investigation can be continued by running the program again, inputting a value $m_2 = m_1 + 2$ to find any/all extrapolation values applicable to other possible factors.

If p is prime, the program will terminate with "No Result" and the following program, (#3), should be used to further test the p associated Mersenne composite for its initial factor.

4.3 Determination of the Initial Factor of a Single Mersenne Composite when the Exponent is Prime, (Program #3).

When p is prime, the associated Mersenne number can be either prime or composite. If composite this program will produce the first factor reasonably quickly. If prime, computation time will be excessively long and therefore this program should be run in parallel with the Lucas-Lehmer test.

```

print("\n\nThis Program finds the First Factor from the Exponent ")
print("when the Exponent is prime, and SHOULD ONLY be used with ")
print("Exponents that are Prime, Otherwise Results may be ")
print("Inconsistent or Factors Missed.\n ")

import time
import math
from mpmath import mp
from mpmath import mpf

p=int(input("\n\nEnter Value of Exponent:- "))
kmax=int(input("Enter the Maximum Value of k:- "))
k=int(input("Enter Start Value of k:- "))
mp.dps=int(p/3)
print("\n")

M=2**p
f=1

```

```

limit=int((2*10**9-1)/(2*p))
fmax=int(2**((p-1)/2)+1)
count=0

start=time.clock()

while f!=fmax:
    f=k*p+1
    q=mpf(M-(f-1)**2)/(2*f)
    if q==int(q):
        print("\n\nM is Composite, the low factor is "),
        print(f)
        break
    elif count==limit:
        print("f so far is, "),
        print(f)
        k +=2
        count =0
    elif f>fmax:
        print("\nM is Prime ")
        break
    elif k>kmax:
        print("\nNo Result, Increase kmax ")
        break
    else:
        k=k+2
        count +=1

finish=time.clock()
run=finish-start
hours=int(run/3600)
run=run-hours*3600
minutes=int(run/60)
seconds=run-minutes*60

print("\n\nRun Time is:- ")
print(hours, minutes, seconds)

input("\n\npress the enter key to exit. ")

```

Table 4.3 – Program #3, To Find the First Factor when p is Prime.

The output of this program is discussed in Section 4.6.3 below.

4.4 For a Single Factor, Determination of the Exponents of the Multiple Associated Mersenne Composites, (Program #4).

This program will, for any particular factor, determine all of the exponents of the applicable Mersenne composites of which it is a factor, up to some selected maximum exponent.

```

print("\nThis Program Finds all the Exponents of Mersenne Composites ")
print("for one Single Factor.\n")

import time
import math
from mpmath import mp
from mpmath import mpf

pmax=int(input("\nEnter Maximum Value of Exponent, pmax:- "))
mp.dps=int(pmax/3)
f=int(input("\nEnter Value of Factor:- "))

p=9
inventory=[]

start=time.clock()

while p<=pmax:
    M=2**p
    n=mpf(2**p-(f-1)**2)/(2*f)
    if n==int(n):
        inventory +=[p]
        p=p+2
    else:
        p=p+2

print("\nThe Exponents are:-\n ")
print(inventory)

finish=time.clock()
run=finish-start
hours=int(run/3600)
run=run-hours*3600
minutes=int(run/60)
seconds=run-minutes*60

print("\n\nRun Time is:- ")
print(hours, minutes, seconds)

input("\n\npress the enter key to exit. ")

```

Table 4.4 - Program #4, Single Factor Determination of Multiple Exponents.

This program will for a single factor print, (on screen), a list of all the applicable exponents.

4.5 For a Single Exponent, Determination of All Factors of the Associated Mersenne Composite.

This program will print, (on screen), for a single exponent all of the factors of the associated Mersenne composite. The program may run for a significant period of time if any factor is a large prime number.

```

print("\n\nTo find all the factors of a single Mersenne Composite This Program ")
print("\n\nShould Only be Used with Mersenne Numbers that are Composite, ")
print("\n\nOtherwise Excessive Run Times May Result. ")

import time
import math
from mpmath import mp
from mpmath import mpf

p=int(input("\nEnter Value of the Exponent:- "))
if p<45:
    d=15
else:
    d=int(p/3)
mp.dps=d
print("\n")

f=7
M=2**p-1
inventory=[]
fmax=int(2**((p-1)/2)+1)

start=time.clock()

while f<=fmax:
    if M==1 or M<f:
        break
    n=mpf(M-f**2)/(2*f)-1
    if n==int(n):
        inventory +=[f]
        M=mpf(M)/f
        print(f),
        fmax=int(M**0.5)
        f +=2
    else:
        f +=2
if M>1:
    M=int(M)
    inventory +=[M]
print("\n\nThe Factors are:-\n ")
print(inventory)

finish=time.clock()
run=finish-start
hours=int(run/3600)
run=run-hours*3600
minutes=int(run/60)
seconds=run-minutes*60

print("\n\nRun Time is:- ")

```

```
print(hours, minutes, seconds)

input("\n\npress the enter key to exit. ")
```

Table 4.5 – Single Exponent Determination of Multiple Factors.

4.6 Examples of Program Application.

This Section provides sample output from all of the above programs. Note that for conciseness, only small values of the exponents are used.

4.6.1 Program #1.

The following is a printout of program #1 with a maximum exponent input of 87 and a start value of 9.

```
To Find the First Factor from the Exponents.

Enter Maximum Value of Exponent, pmax:- 87
Enter Start Value of Exponent. Must be less than or equal to pmax:- 9

('p is', 9) ('f is', 7)
('p is', 11) ('f is', 23)
('p is', 15) ('f is', 7)
('p is', 21) ('f is', 7)
('p is', 23) ('f is', 47)
('p is', 25) ('f is', 31)
('p is', 27) ('f is', 7)
('p is', 29) ('f is', 233)
('p is', 33) ('f is', 7)
('p is', 35) ('f is', 31)
('p is', 37) ('f is', 223)
('p is', 39) ('f is', 7)
('p is', 41) ('f is', 13367)
('p is', 43) ('f is', 431)
('p is', 45) ('f is', 7)
('p is', 47) ('f is', 2351)
('p is', 49) ('f is', 127)
('p is', 51) ('f is', 7)
('p is', 53) ('f is', 6361)
('p is', 55) ('f is', 23)
('p is', 57) ('f is', 7)
('p is', 59) ('f is', 179951)
('p is', 61) and 2^p-1, is Possibly Prime.
('p is', 63) ('f is', 7)
('p is', 65) ('f is', 31)
('p is', 67) and 2^p-1, is Possibly Prime.
('p is', 69) ('f is', 7)
('p is', 71) ('f is', 228479)
('p is', 73) ('f is', 439)
('p is', 75) ('f is', 7)
('p is', 77) ('f is', 23)
('p is', 79) ('f is', 2687)
('p is', 81) ('f is', 7)
```

(p is', 83) ('f is', 167)
 ('p is', 85) ('f is', 31)
 ('p is', 87) ('f is', 7)

Run Time is:-
 (0, 3, 54.69)

Press the enter key to exit.

Table 4.6 – Sample Output of Program #1.

The printout shows the low factor for all composite Mersenne numbers up to $p = 87$. Note that the Mersenne numbers with $p = 61$ and $p = 67$ are indicated as possibly prime. $M_{61} = 2^{61} - 1$ is prime but M_{67} is composite, ($f = 193,707,721$). Both numbers should therefore be tested with program #3 as is shown below for M_{67} .

Note that where a value of p is missing from the above list, i.e. 13, 31 etc, the associated Mersenne number is prime.

To determine the first factor of a large range of exponents, to minimise run times, it may be preferred to run in batches of say, 1,000. This will permit a large range to be explored over an extended period of time.

There are three factors in the above printout that are themselves Mersenne numbers, $7 = 2^3 - 1$, $31 = 2^5 - 1$ and $127 = 2^7 - 1$. Consequently, the additional exponents for which these numbers are also factors are, (as per Case (ii) in Section 3.2), (only odd exponents shown).

<i>f</i>	Additional Exponents
7	15, 21, 27, 33 ----- $p_i, p_i + 6$
31	35, 45, 55, 65----- $p_i, p_i + 10$
127	63, 77, 91, 105----- $p_i, p_i + 14$

Table 4.7 – Exponents of Mersenne Numbers for which *f* is a Factor, when *f* is Also a Mersenne Number.

For the other factors that are not themselves Mersenne numbers, the additional exponents for which these numbers are also factors, are, (as per Case (i) in Section 3.2)

<i>f</i>	Additional Exponents
23	33, 55, 77, 99----- $p_i, p_i + 22$
47	69, 115, 161, 207----- $p_i, p_i + 46$
233	87, 145, 203, 261----- $p_i, p_i + 58$
223	111, 185, 259, 333----- $p_i, p_i + 74$
13367	123, 205, 287, 369----- $p_i, p_i + 82$
431	129, 215, 301, 387----- $p_i, p_i + 86$
2351	141, 235, 329, 423----- $p_i, p_i + 94$
69431	159, 265, 371, 477----- $p_i, p_i + 106$
179951	177, 295, 413, 531----- $p_i, p_i + 118$
228479	213, 355, 497, 639----- $p_i, p_i + 142$
439	219, 365, 511, 657----- $p_i, p_i + 146$

2687	237, 395, 553, 711----- $p_i, p_i + 158$
167	249, 415, 581, 747----- $p_i, p_i + 166$

Table 4.8 - Exponents of Mersenne Numbers for which f is a Factor, when f is Not Also a Mersenne Number.

The above two tables are of course constructed via the results of Section 3.2. As can be seen there is a great deal of duplication and overlap. Also, for some factors there are exponents that are lower than those shown. This will become clearer, and the lower exponents identified, in the examples for program #2 below.

4.6.2 Program #2.

This program will produce the exponent extrapolation value of p applicable to the lowest value of the iteration parameter m . This is important in its interpretation. A printout from program #2 is shown below for exponents 77 and 8191.

This Program Determines the Extrapolation Values,
for any Particular Mersenne Number Exponent.

Enter Value of Exponent:- 77
Enter Value of m, Must be Odd and Minimum is 3:- 3

p is Composite.

The p Extrapolation Value is 14

Run Time is:-
(0, 0, 0.01)

Press the enter key to exit.

Table 4.9 – Sample Output of Program #2, ($p = 77$).

This Program Determines the Extrapolation Values,
for any Particular Mersenne Number Exponent.

Enter Value of Exponent:- 8191
Enter Value of m, Must be Odd and Minimum is 3:- 3

No Result, p is Prime

Run Time is:-
(0, 0, 0.39)

Press the enter key to exit.

Table 4.10 – Sample Output of Program #2, ($p = 8191$).

To interpret the exponent extrapolation value of 14 in Table 4.9 above, note that it corresponds to the factor of M_{77} of 127 in Table 4.7, which is $2^7 - 1$, itself a Mersenne number. This extrapolation value should therefore be applied so as to produce the entries in Table 4.7, i.e. $77 + 14 = 91$ etc. Also note that extrapolation

values can be applied down as well as up. For instance, applying the value of 14 down produces the following values of p :- 63, 49, 35, 21, and 7, the Mersenne numbers of which all have 127 as a factor, (the Mersenne number with the exponent value 7 is of course also 127 and so this is a trivial result).

To demonstrate more clearly how program #2 works, truth tables are shown in Appendix C for exponents 55 and 77.

4.6.3 Program #3.

In cases where p is confirmed prime, it is necessary to test M_p using program #3. In this program the iteration value is that of (3.21) because only a single prime exponent is being tested. This minimises computation time.

However, if M_p itself is prime, this program will take an exceptionally long time to run, and it should therefore be run in parallel with the Lucas-Lehmer test.

If M_p is composite, the program will terminate with its first factor about 50% of the time. Note that this is the case when $k/2, (= (f - 1)/2p)$, is less than $p - 2$. Even if this is not so, the program will always provide the first factor, whereas the Lucas-Lehmer test provides no factors. If M_p is prime the Lucas-Lehmer test will terminate first.

The printouts are the result of testing M_{67} and M_{541} .

This Program finds the First Factor from the Exponent when the Exponent is prime, and SHOULD ONLY be used with Exponents that are Prime, Otherwise Results may be Inconsistent or Factors Missed.

Enter Value of Exponent. 67
 Enter the Maximum Value of k:- 10000000
 Enter Start Value of k:- 100000

M is Composite, the low factor is:- 193707721

Run Time is:-
 (0, 2, 22.46)

press the enter key to exit.

Table 4.11 – Program #3, The First Factor of M_{67} .

In this case the test was started with a maximum value of k specified at 1,000,000 and a start value of 2. This gave a "No Result" and asked for $kmax$ to be increased. A Value of 10,000,000 for $kmax$ with a start value of 1,000,000, (the previous $kmax$), then gave the above result.

This Program finds the First Factor from the Exponent when the Exponent is prime, and SHOULD ONLY be used with Exponents that are Prime, Otherwise Results may be Inconsistent or Factors Missed.

Enter Value of Exponent:- 541
 Enter the Maximum Value of k:- 100000000
 Enter Start Value of k:- 2

f so far is, 2000000179

```
f so far is, 4000000357
M is Composite, the low factor is:- 4312790327
Run Time is:-
(0, 8, 19.66)
press the enter key to exit.
```

Table 4.12 – Program #3, The First Factor of M_{541} .

Note that the "f so far" comment is just to indicate the extent to which the calculation has proceeded.

4.6.4 Program #4.

The following is a printout of all the exponents of Mersenne numbers, up to a maximum of 1029, for which 23 is a factor.

```
This Program Finds all the Exponents of Mersenne Composites
for one Single Factor.

Enter Maximum Value of Exponent, pmax:- 1029

Enter Value of Factor:- 23

The Exponents are:

[11, 33, 55, 77, 99, 121, 143, 165, 187, 209, 231, 253, 275, 297, 319, 341, 363, 385,
407, 429, 451, 473, 495, 517, 539, 561, 583, 605, 627, 649, 671, 693, 715, 737, 759,
781, 803, 825, 847, 869, 891, 913, 935, 957, 979, 1001, 1023]

Run Time is:-
(0, 0, 0.08)

press the enter key to exit.
```

**Table 4.13 – All Exponents of Mersenne Numbers Up to $p = 1029$,
for which 23 is a Factor.**

Note that while the above information could also be determined using programs # 1 and 2, this program is quicker and requires no searching or interpolation. Also note that the exponent extrapolation value is 22.

4.6.5 Program #5.

The following is a printout of all the factors of the single Mersenne composite with exponent $p = 105$.

```
To find all the factors of a single Mersenne Composite. This Program Should Only
be Used with Mersenne Numbers that are Composite, Otherwise Excessive Run
Times May Result.

Enter Value of the Exponent:- 105
```

7 31 71 127 151 337 29191 106681 122921

The Factors are:-

[7, 31, 71, 127, 151, 337, 29191, 106681, 122921, 1064287]

Run Time is:-

(0, 0, 10.81)

Press the enter key to exit.

Table 4.14 – All Factors of the Mersenne Number M_{105} .

Note that the top list of factors are printed out as they are found as an indication of progress. Also, note that where p is large, (and for some small values), run time will be long because the higher factor(s) will be very large.

5.0 Conclusions

The system enabled in this paper has provided the means to derive all the factors of any Mersenne number, regardless of the size of its exponent. And it is to that end that the five Python programs have also been provided.

However, to an experienced Python programmer, the programs provided here will be considered somewhat "clunky". While, for Mersenne numbers with small exponents and/or "small" prime factors, they are perfectly adequate, for some numbers with extremely large exponents and/or extremely large prime factors, they will take an unacceptably long time to run. No doubt however, they could be improved significantly to reduce run time when factoring such numbers, but would still be inadequately slow for Mersenne numbers with many thousands/millions of digits. Consequently, this could only be avoided by coding the programs in a low level language such as Assembler.¹

The other main problem with these programs, irrespective of the computer language in which they were coded, occurs when attempting to test a large Mersenne number that was prime. Run times in this case would be even longer and the only way to test such numbers at the moment is to use the Lucas-Lehmer test as embodied in the software used in the GIMPS project. Consequently, for aspiring Mersenne prime investigators, joining the GIMPS project is thoroughly recommended.

APPENDIX A.

Analytical Factorisation of Mersenne Composites when p is Composite.

A.1 p is Even.

When p is even, factorisation is trivial. If

$$M_p = 2^p - 1 \quad (\text{A.1})$$

then

¹ A task not at the moment within the capabilities of the author.

$$M_p = \left(2^{p/2} - 1\right) \left(2^{p/2} + 1\right) \quad (\text{A.2})$$

A.2 p is Odd and Composite.

When p is odd and composite and with

$$M_p = 2^p - 1 \quad (\text{A.3})$$

Where $p = a \times b$, then

$$\begin{aligned} M_p &= (2^a - 1)2^{(p-a)} + 2^{(p-a)} - 1 \\ &= (2^a - 1)2^{(p-a)} + (2^a - 1)(2^{(p-2a)}) - 1 \\ &= (2^a - 1)2^{(p-a)} + (2^a - 1)(2^{(p-2a)}) + \dots + (2^a - 1)(2^{(p-qa)}) - (2^a - 1) \\ &= (2^a - 1) \sum_{q=1}^{(b-1)} (2^{(p-qa)}) \end{aligned} \quad (\text{A.4})$$

This gives the first and second factors which may have further sub-factors as with M_{55} and M_{77} . When p is extremely large, determination of all the factors analytically will however be very slow and laborious.

APPENDIX B.

Fractional Values of k in Factors of Mersenne Numbers when Factors are Expressed as $f = 2kp + 1$.

When the factor of a Mersenne composite is expressed as

$$f = 2kp + 1 \quad (\text{B.1})$$

if $(f - 1)$ is not an exact multiple of p , the value of k will be fractional. To illustrate this the table below shows the value of k for all the factors found in the printout for program #1.

p	f	$k = (f - 1)/2p$
9	7	0.4444
11	23	1
15	7	0.2666
21	7	0.1905
23	47	1
25	31	0.6000
27	7	0.1481
29	233	19
33	7	0.1212
35	31	0.4286
37	223	3
39	7	0.1026
41	13367	163
43	431	5

45	7	0.0889
47	2351	25
49	127	1.2857
51	7	0.0784
53	69431	665
55	23	0.2000
57	7	0.0702
59	179951	1525
63	7	0.0635
65	31	0.2308
69	7	0.0580
71	228479	1609
73	439	3
75	7	0.5333
77	23	0.1428
79	2687	17
81	7	0.0494
83	167	1
85	31	0.1765.
87	7	0.0460

Table B.1 – Values of k for Results from Program #1 Example.

Thus in less than 50% of these cases is k an integer. However, it is also noted that in most, if not all cases, k is an integer for the second factor.

APPENDIX C.

Truth Tables for Program #2.

The following are truth tables for program #2 for exponents 55 and 77.

p	m	$q = p/m$	$q < 2$	$q = \text{int}(q)$	Action.
55	3	8.3	No	No	$m = m+2$
	5	11.0	No	Yes	Extrap. = $2m = 10$
	7	7.8	No	No	$m = m+2$
	9	6.1	No	No	$m = m+2$
	11	5.0	No	Yes	Extrap. = $2m = 22$

Table C.1 – Truth Table for $p = 55$ Using Program #2.

p	m	$q=p/m$	$q < 2$	$q = \text{int}(q)$	Action.
77	3	25.6	No	No	$m = m+2$
	5	15.4	No	No	$m = m+2$
	7	11.0	No	Yes	Extrap. = $2m = 14$
	9	8.6	No	No	$m = m+2$
	11	7.0	No	Yes	Extrap. = $2m = 22$

Table C.2 – Truth Table for $p = 77$ Using Program #2.

Here it is seen that the extrapolation values in the above two tables, tie up precisely with those in Tables 4.7 and 4.8.

REFERENCES.

- [1] P.G.Bass, *An Algorithmic Version of the Sieve of Eratosthenes Plus Other Solutions Associated With Prime Numbers. (Version 2)*, [www.relativitydomains .com](http://www.relativitydomains.com).
- [2] The GIMPS Project, *Great Internet Mersenne Prime Search*, www.Mersenne.org.