Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 19, Number 2 (2023), pp. 717-726
©Research India Publications
http://www.ripublication.com/gjpam.htm

Superabundant Numbers and Maxima and Minima in Subsets of the Natural Numbers

Darrell Cox[1]

[1]*Department of Mathematics, Grayson County College, United States*
E-mail: darkencox273@gmail.com

Abstract

Euler's totient function is used to define the subsets of the natural numbers. "Primes" in these subsets are associated with the maxima. "Staircases" of primes can be generated just as for the usual primes.

Keywords: Euler's totient function, superabundant numbers, staircase of primes, Riemann hypothesis

1. INTRODUCTION

Let $\varphi(n)$ denote Euler's totient function and

$$r(n) = \frac{\prod_{i|n}(1 + \frac{1}{\varphi(i)})}{\prod_{i|n}(1 + \frac{1}{i})}. \tag{1}$$

Let N_f denote a subset of the natural numbers where $r(n) < f$ and f is a rational number greater than 1. Let $\sigma(n)$ denote the sum of divisors function. The maxima and minima are defined to be relative to $n/\sigma(n)$. A natural number is said to be superabundant if $\sigma(m)/m < \sigma(n)/n$ for all $m < n$. For f less than about 1.4, the superabundant numbers are the minima of N_f. For f greater than or equal to 1.4, the first few superabundant numbers are missing. Determining the maxima in N_f is more involved. In general, the n values in N_f are even so there are no primes. The "primes" in N_f are associated with the maxima. For example, for $N_{2.0}$ the primes are powers of two times a subset of the usual primes.

2. $f = 2.0$

A plot of N_f for $n \leq 3000$ (here n denotes the natural numbers used to derive N_f) is

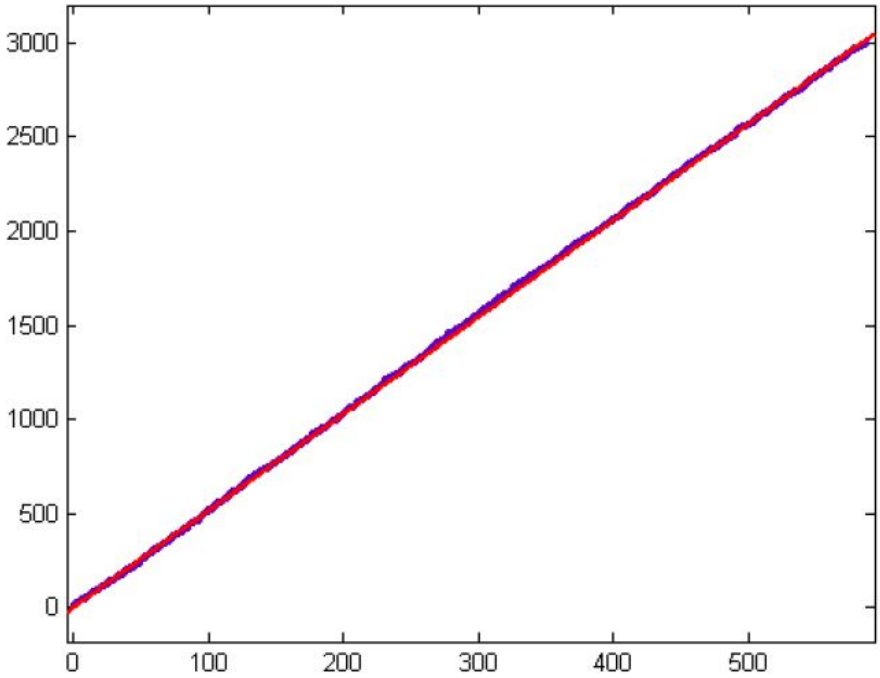

Figure 1:

For a linear least-squares fit of the curve, $p_1 = 5.139$ with a 95% confidence interval of (5.134, 5.143), $p_2 = -0.5828$ with a 95% confidence interval of (-2.061, 0.8957), SSE=$4.852 \cdot 10^4$, R-squared=0.9999, and RMSE=9.107. In general, the values are almost linear.

There are 587 values. The first few n values are 12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60,.... The superabundant numbers less than 1500000 are 2, 4, 6, 12, 24, 36, 48, 60, 120, 180, 240, 360, 720, 840, 1260, 1680, 2520, 5040, 10080, 15120, 25200, 27720, 55440, 110880, 166320, 277200, 332640, 554400, 665280, 720720, and 1441440. Other than 2, 4, and 6, the superabundant numbers have been confirmed to among these n values for a natural number upper bound of 1500000.

A plot of $n/\sigma(n)$ for $n \leq 3000$ is

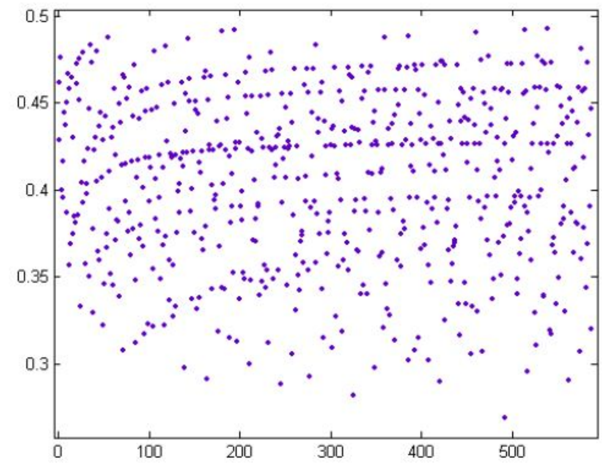

Figure 2:

The n values of the first few lower bounds of these ratios are 12, 24, 36, 48, 60, 120, 180, 240, 360, 720,. . .. These are superabundant numbers.

A plot of the upper bounds of $n/\sigma(n)$ for the 98698 n values corresponding to a natural number upper bound of 500000 is

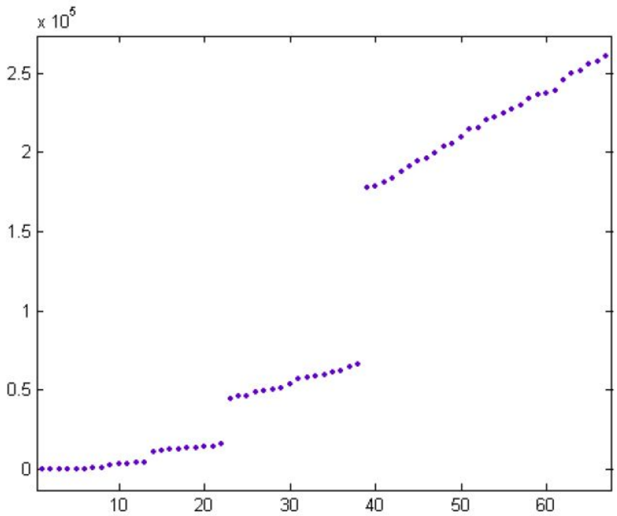

Figure 3:

The first few n values are $2^2 \cdot 3$, $2 \cdot 3^2$, $2^2 \cdot 5$, $2 \cdot 3 \cdot 23$, $2 \cdot 3 \cdot 29$, $2^4 \cdot 17$, $2^5 \cdot 29$, $2^5 \cdot 31$,

$2^6 \cdot 43, 2^6 \cdot 47, 2^6 \cdot 47, 2^6 \cdot 53, 2^6 \cdot 59, 2^6 \cdot 61$. Other than $2 \cdot 3^2$, $2 \cdot 3 \cdot 23$, and $2 \cdot 3 \cdot 29$, a sequence of primes is generated for a fixed power of 2 and the ending prime is almost equal to the power of 2. For 2^2 the ending prime is 5, for 2^4 the ending prime is 17, for 2^5 the ending prime is 31, for 2^6 the ending prime is 61, for 2^7 the ending prime is 127, for 2^8 the ending prime is 257, and for 2^9 the ending prime is 509. A plot of the logarithms of these n values excluding the first five is

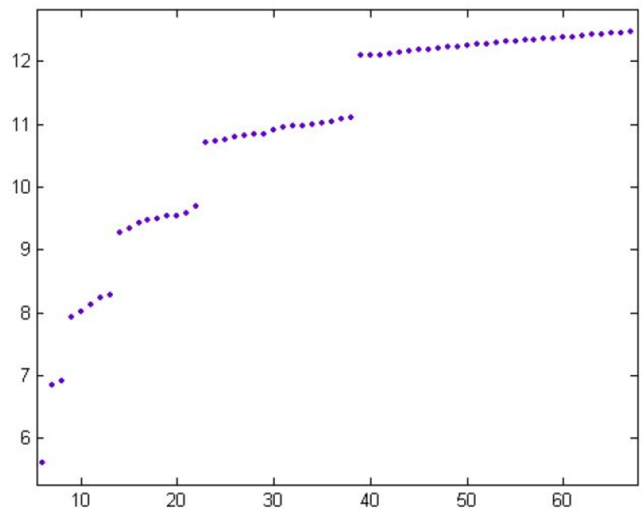

Figure 4:

The logarithms of the n values are locally almost straight. A plot of the logarithms of the n values corresponding to a natural number upper bound of 1500000 is

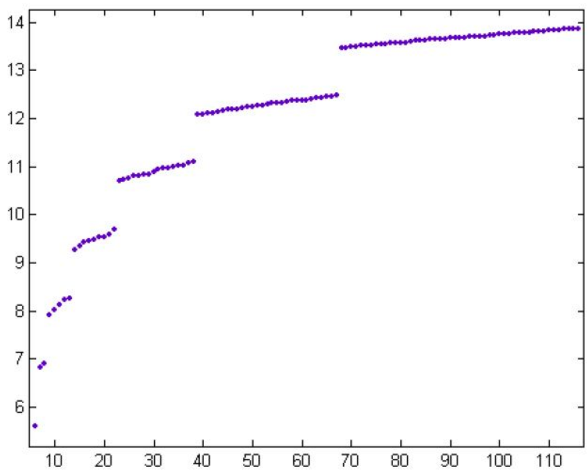

Figure 5:

For 2^{10}, the ending prime is 1021.

3. $f = 2.5$

For $180 \leq n \leq 1500000$, the maxima are at $2^2 \cdot 3^2 \cdot 5$, $2^6 \cdot 3$, $2 \cdot 3^2 \cdot 11$, $2 \cdot 3^3 \cdot 17$, $2^3 \cdot 5^3$, $2 \cdot 3^3 \cdot 17$, $2 \cdot 3^3 \cdot 19$, $2 \cdot 3^3 \cdot 23$, $2 \cdot 3^4 \cdot 29$, $2 \cdot 3^4 \cdot 31$, $2 \cdot 3^5 \cdot 37$, $2 \cdot 3^6 \cdot 41$, $2 \cdot 3^7 \cdot 43$. A plot of the logarithms of the n values that are 2 times a power of 3 times 11, 17, 19, 23, 29, 31, 37, 41, and 43 respectively is

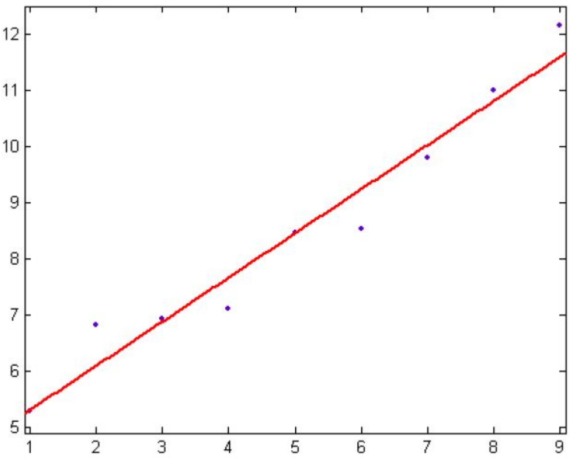

Figure 6:

4. $f = 3.0$

For $50 \leq n \leq 1500000$, the first few maxima are at $2^2 \cdot 3 \cdot 5$, $2^3 \cdot 3^2$, $2^2 \cdot 3 \cdot 7$, $2 \cdot 3^2 \cdot 5$, $2^2 \cdot 3^3$, $2^2 \cdot 3 \cdot 11$ and $2^2 \cdot 3 \cdot 13$. Disregarding these values, the maxima are at $2^3 \cdot 3$ times the primes starting with 197. A plot of these values is

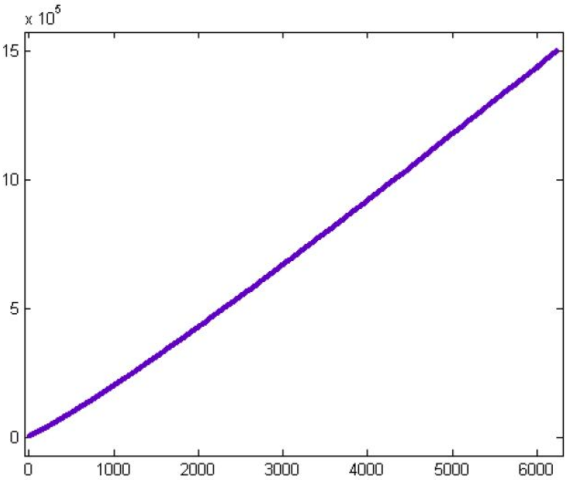

Figure 7:

The last value is $2^3 \cdot 3 \cdot 62497$.

5. $f = 3.5$

For $70 \leq n \leq 1500000$, the first three maxima are at $2^3 \cdot 3^2$, $2^5 \cdot 3$, and $2^2 \cdot 3^3$. Thereafter, the maxima are at $2^4 \cdot 3 \cdot 281$, $2^4 \cdot 3 \cdot 283$, $2^4 \cdot 3 \cdot 293, \ldots, 2^4 \cdot 3 \cdot 31249$. All the primes in this range are included.

6. $f = 4.0$

For $300 \leq n \leq 1500000$, the maxima are at $2^2 \cdot 3 \cdot 5^2$, $2^4 \cdot 3^3$, $2^4 \cdot 3 \cdot 11$, $2^4 \cdot 3 \cdot 13$, $2^5 \cdot 3 \cdot 17$, $2^5 \cdot 3 \cdot 19$, $2^5 \cdot 3 \cdot 23$, $2^6 \cdot 3 \cdot 29$, $2^6 \cdot 3 \cdot 31$, $2^6 \cdot 3 \cdot 37$, $2^7 \cdot 3 \cdot 47$, $2^8 \cdot 3 \cdot 53$, $2^8 \cdot 3 \cdot 59$, $2^8 \cdot 3 \cdot 61$, $2^9 \cdot 3 \cdot 71$, and $2^{11} \cdot 3 \cdot 79$. In this case, the powers of 2 increase in a somewhat erratic manner and some of the primes (41, 43, and 73) are missing. A plot of the logarithms of the values (excluding the first two) is

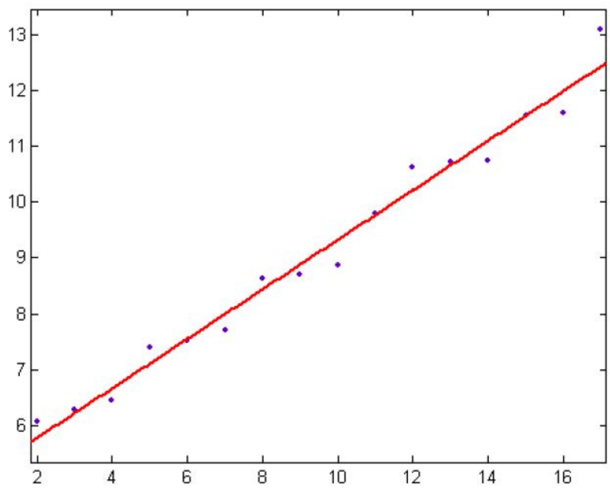

Figure 8:

7. CONCLUSION

The results for the $f = 3.0$ and $f = 3.5$ cases are unexpected in that colossally abundant numbers are usually used to derive conditions equivalent to the Riemann Hypothesis. See Lagarias [1] for example.

8. METHODS

The following C code computes N_j and finds maxima and minima. The prime look-up table contains the primes less than 1500000.

```
//
```

```
// compute Mobius function
//
#include <math.h>
#include <stdio.h>
#include "table5.h" // prime look-up table
int mobius(unsigned int a, unsigned int *table, unsigned int tsize) {
unsigned int i,count,p;
if (a==1)
    return(1);
count=0;
for (i=0; i<tsize; i++) {
    p=table[i];
    if (p>a)
        break;
    if (a==(a/p)*p) {
        a=a/p;
        if (a==(a/p)*p)
            return(0);
        count=count+1;
        if (a==1)
            break;
        }
    }
if ((count&1)==0)
    return(1);
else
    return(-1);
}
//
// compute Euler's phi function
//
int mobius(unsigned int a, unsigned int *t, unsigned int tsize);
unsigned int nueuler(unsigned int n, unsigned int *table,
unsigned int tsize) {
unsigned int d;
int sum;
if (n==1)
    return(1);
sum=0;
```

```
for (d=1; d<=n; d++) {
    if (n==(n/d)*d)
        sum=sum+(n/d)*mobius(d, table, tsize);
    }
return((unsigned int)sum);
}
//
// Products of 1 plus reciprocals of totient function
// divided by products of 1 plus the reciprocal of n
//
unsigned int nueuler(unsigned int a, unsigned int *table, unsigned int tsize);
unsigned int max=100000;
double maxr=2.0; // normally set to 2.0
unsigned int out=0; // if set to 0, output n, sigma(n), and ratios
// if set to 1, only output n
// if set to 2, output n and sigma(n)
// if set to 3, output maxima
// if set to 4, output minima
unsigned int offset=0;
unsigned int tsize=114155; // size of prime look-up table
void main() {
unsigned int n,i,temp,count,sum1;
double sum,sum2,minnr,maxxr,r;
FILE *Outfp;
Outfp = fopen("cox8g.dat","w");
if (max>1500000) {
    printf("max too large \n");
    return;
    }
minnr=999999.0;
maxxr=0.0;
count=0;
for (n=1; n<=max; n++) {
    sum=1.0;
    sum1=0;
    sum2=1.0;
    for (i=1; i<=n; i++) {
        if (n==(n/i)*i) {
            temp=nueuler(i,table,tsize);
```

```
            sum=sum*(1.0+1.0/(double)temp);
            sum2=sum2*(1.0+1.0/(double)i);
            sum1=sum1+i;
            }
        }
    if ((sum/sum2)>maxr) {
        count=count+1;
        if (out==0)
            fprintf(Outfp," %d %.10llf %.10llf %.10llf \n",n,sum,sum2,sum/sum2);
        if (out==1)
            fprintf(Outfp," %d, \n",n);
        if (out==2)
            fprintf(Outfp," %d, %d, \n",n,sum1);
        r=(double)n/(double)sum1;
        if ((r>maxxr)&&(n>=offset)) {
            maxxr=r;
            if (out==3) {
                fprintf(Outfp," %d, \n",n);
                printf(" %d, \n",n);
                }
            }
        if (r<minnr) {
            minnr=r;
            if (out==4) {
                fprintf(Outfp," %d, \n",n);
                printf(" %d, \n",n);
                }
            }
        }
    if ((out==0)&&(n==(n/1000)*1000))
        printf(" %d %d %.10llf %.10llf \n",n,sum1,minnr,maxxr);
    }
printf("count=%d \n",count);
fclose(Outfp);
return;
}
```

REFERENCES

[1] J. C. Lagarias, An Elementary Problem Equivalent to the Riemann Hypothesis, arXiv: math/0008177v2 [math.NT] 6 May 2001