COMPLEX SPACE FACTORIZATION

FRED VIOLE[f]

ABSTRACT. We present a visualization of the complex space of possible factors to a real number N and the area covered through different factorization methods (trial division & Fermat's method). We propose a method of restricting each method to its optimal location in the complex space, while retaining their ability to be sieved. We further note an observation to one of Fermat's methods that drastically reduces this complex space. However, at this point we are uncertain of its full implications.

1. COMPLEX SPACE

Each triangle in the following images represents a complex number. The complex number is of the form $(a + bi)$. From our work on the definition of i[♦], we were able to assign real numbers to each complex number. For example, the complex number $(12 + i)$ represents the real numbers$[11,13]$.

- The blue complex numbers represent real numbers, that when multiplied, are $< N$.
- The red complex numbers represent real numbers, that when multiplied, are $> N$.

The factors of N reside in the complex number along the factor strip where the blue and red converge. The green triangle represents the complex number for the factors of N.

[f] Corresponding author: fred.viole@gmail.com

[♦] Working paper available for download: https://www.scribd.com/doc/279451210/i

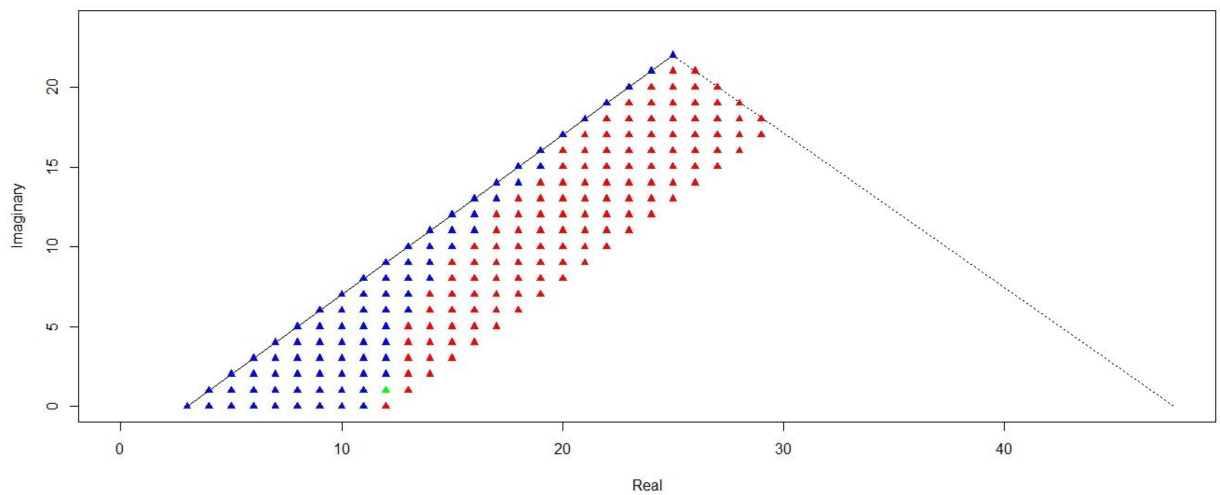

Figure 1. Full complex space of possible factors for $N = 143$. Complex number $(12 + i)$ in green representing the factors $[11, 13]$ for $N = 143$.

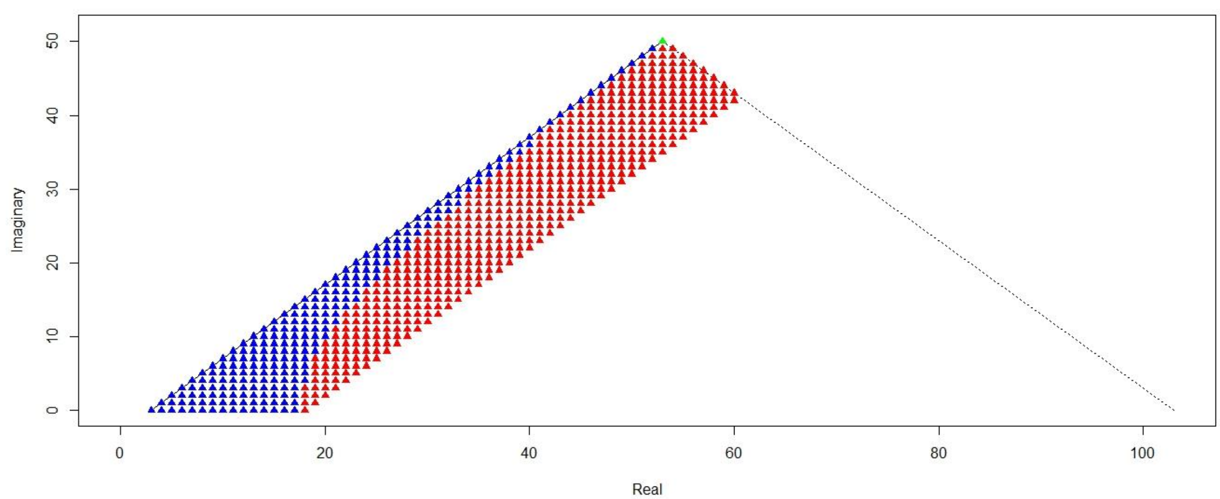

Figure 2. Full complex space of possible factors for $N = 309$. Complex number $(53 + 50i)$ in green representing the factors $[3, 103]$ for $N = 309$.

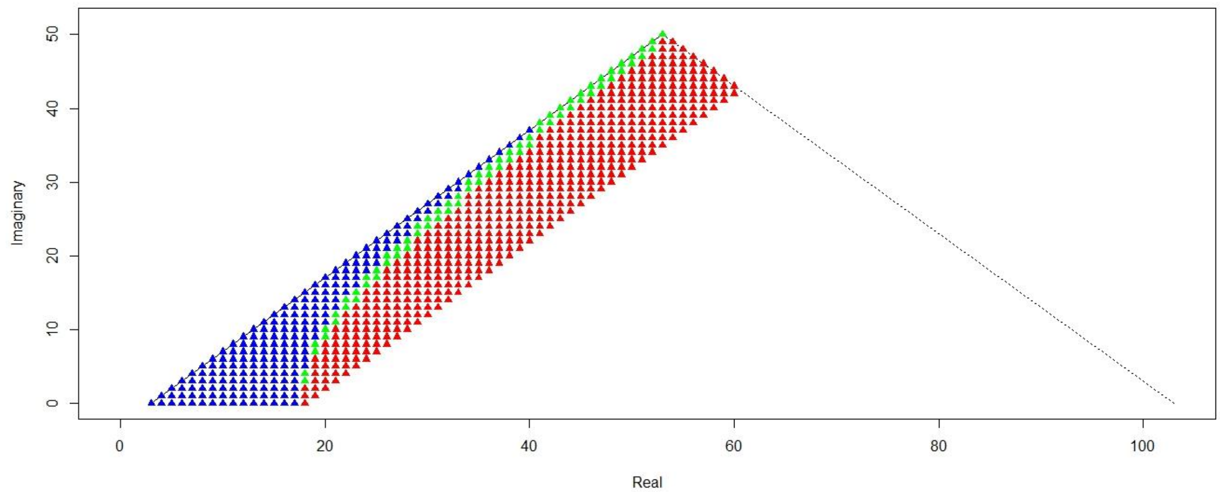

Figure 3. Full complex space of possible factors for $N = 309$. Factor strip highlighted in green.

2. Trial Division

Figures 4 and 5 below illustrate the area of the complex space covered by trial division. We also note the efficiency of only dividing N by the prime numbers less than $\sqrt{N}$. This efficiency is fully realized with a known distribution of prime numbers or a deterministic primality test.

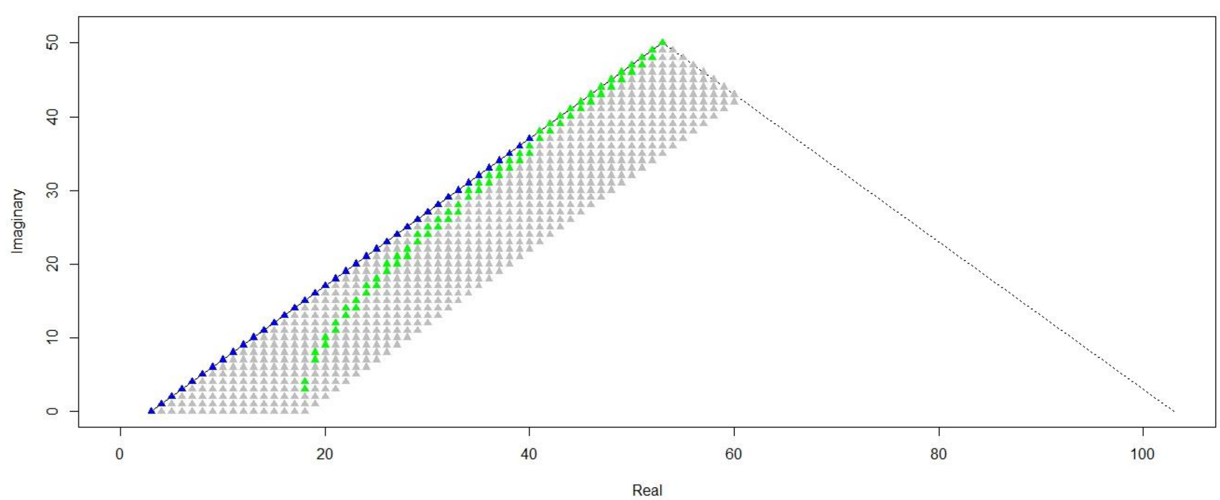

Figure 4. The area of the complex space covered by trial division. $\frac{N}{3}$ highlighted. Factor strip highlighted in green.

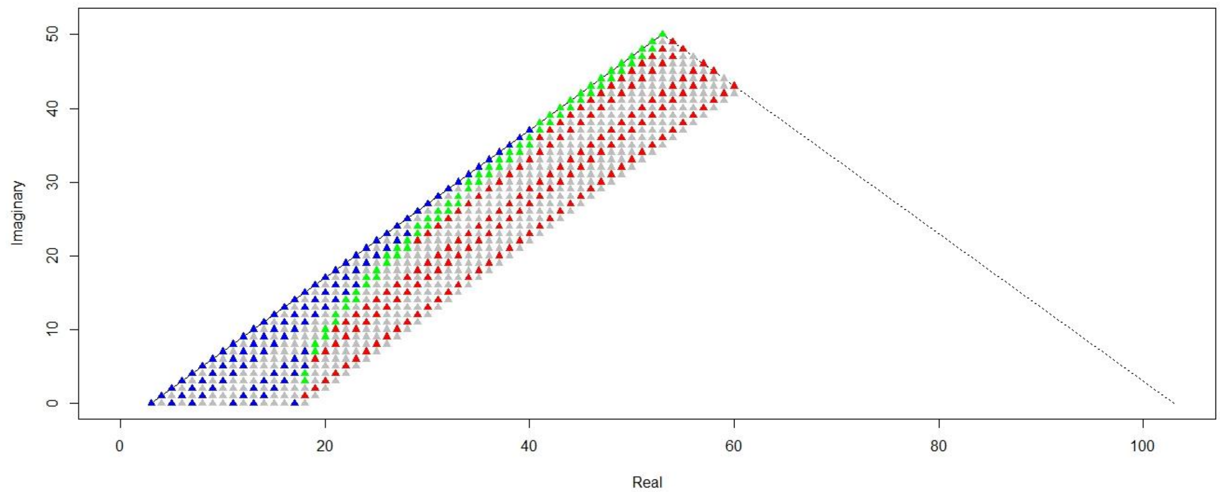

Figure 5. The area of the complex space covered by dividing N by all prime numbers less than $\sqrt{N}$. $\frac{N}{3}, \frac{N}{5}, \frac{N}{7}, \frac{N}{11}, \frac{N}{13}, \frac{N}{17}$ highlighted. Factor strip highlighted in green. Note the efficiency for smaller numbers with trial division.

3. Fermat's Method

Fermat's method involves the difference of squares such that $N = a^2 - b^2$. By rearranging the terms we are left with a vertical or horizontal scan of the complex space.

3.1 Vertical Fermat:

The vertical Fermat scan involves the real part of the complex number, a from $(a + bi)$.

$$\boldsymbol{b^2 = a^2 - N}$$

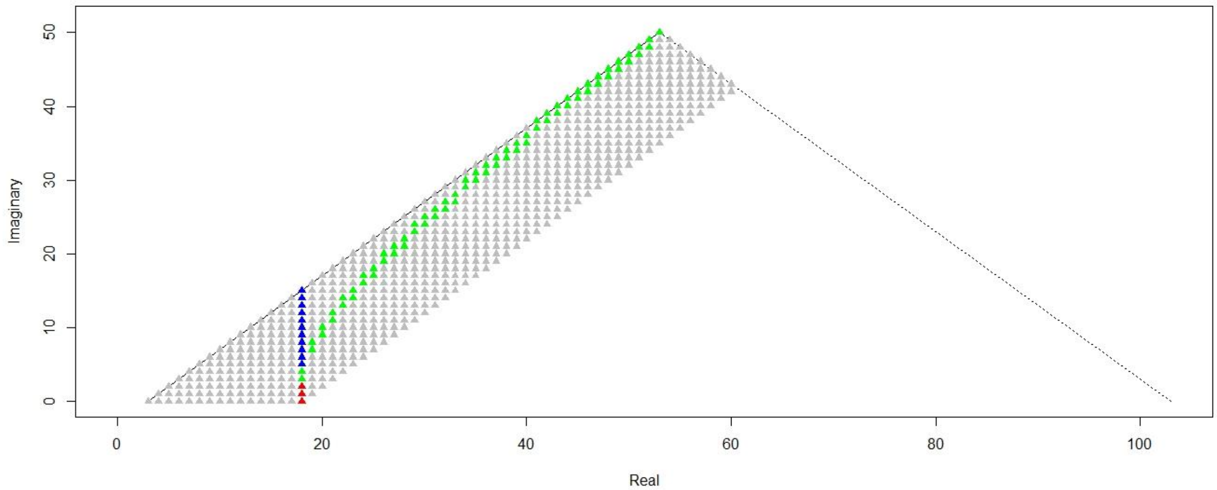

Figure 6. The area of the complex space covered by using Fermat's vertical method starting from $\lceil\sqrt{N}\rceil$. Factor strip highlighted in green.

3.2 Horizontal Fermat:

The horizontal Fermat scan involves the imaginary part of the complex number, b from $(a + bi)$.

$$\boldsymbol{a^2 = N + b^2}$$

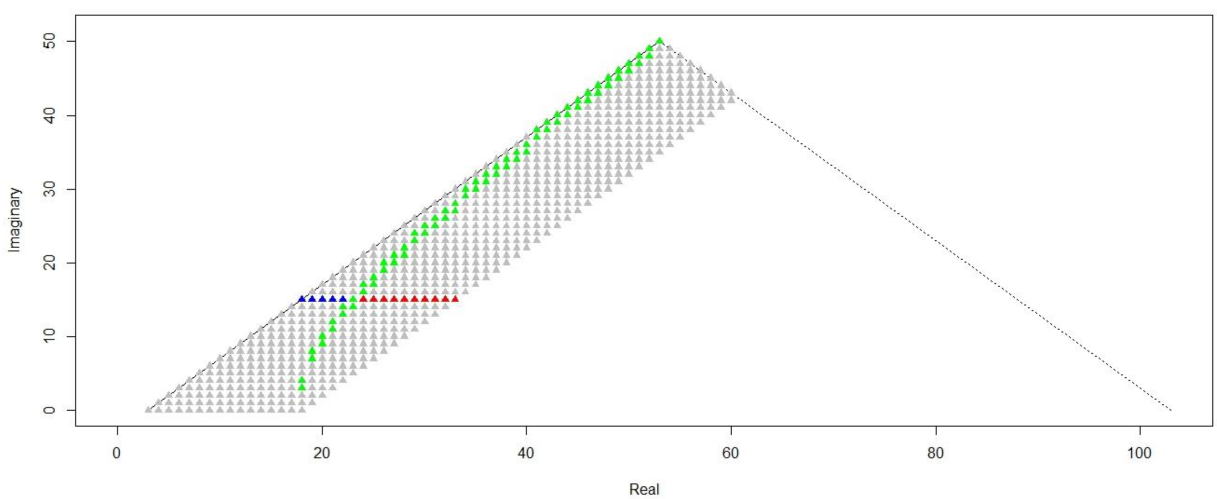

Figure 7. The area of the complex space covered by using Fermat's horizontal method starting from $\lceil\sqrt{N}\rceil - 3$. Factor strip highlighted in green.

4. Combined Method

Given the curvature of the strip from Figure 3, we can see that each of these methods will be most efficient for different parts of the complex space. For instance,

- Vertical Fermat will be most effective from $\lceil\sqrt{N}\rceil$ where the strip is almost vertical.
- Trial division will be most effective where the strip is almost 45° which occurs for smaller real numbers.
- Horizontal Fermat will be most effective in the middle, descending towards the area where it is most effective from $\lceil\sqrt{N}\rceil - 3$.

We can then propose assembling all three methods and running simultaneously as illustrated in Figure 8.

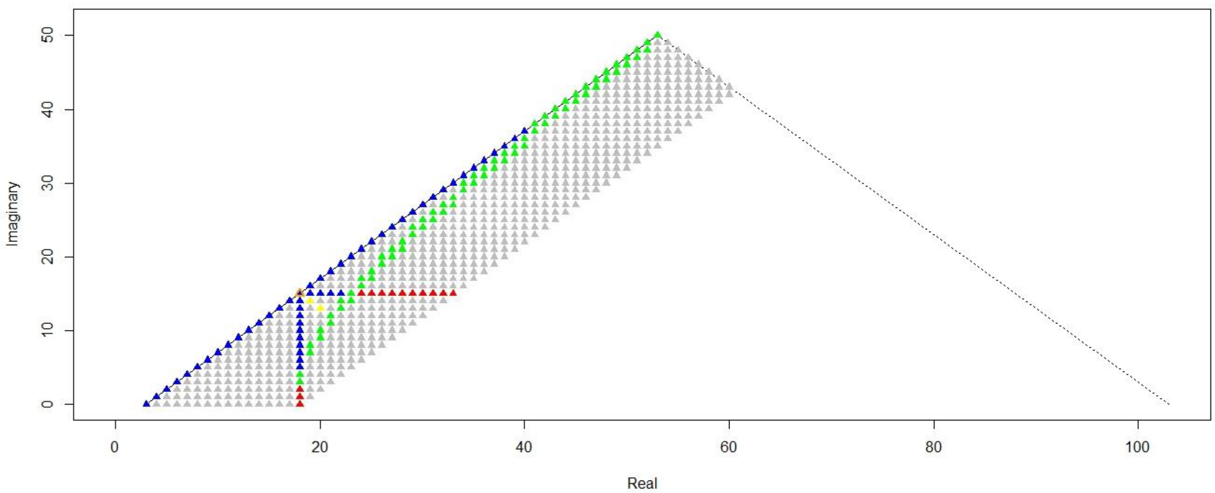

Figure 8. The area of the complex space covered by using all methods starting from the key complex number $\left(\lceil\sqrt{N}\rceil + \left(\lceil\sqrt{N}\rceil - 3\right)i\right)$ highlighted in orange. Factor strip highlighted in green.

We can traverse down the yellow path of complex numbers with each iteration, collapsing the 90° bracket from Fermat's two methods onto the lower third of the factor strip, while the trial division with smaller numbers scans the upper portion of the strip.

We stop when the key complex number is red, representing real numbers too small to be factors of N. Figure 9 illustrates this condition.

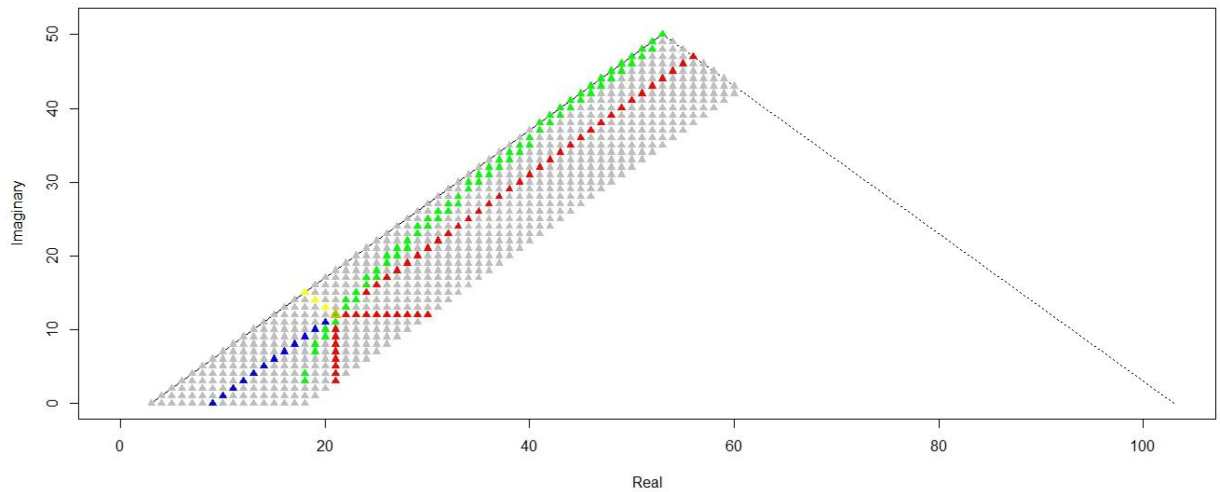

Figure 8. The stopping point for checking the complex space using all three methods simultaneously.

The benefits from this method are:

- *Deterministic*. It will find a factor every time, as opposed to the probabilistic properties of other methods such as Pollard's rho.
- *Parallel Processing*. Since we know the length of the yellow path, it can be divided and separate processors can be used to travel both up & down the path.
- *Sieves*. To reduce the computational burden any of the methods might have, we can translate the current complex number through its real number sieves such as primality for trial division and the Fermat sieves[1] for both a^2 and b^2.

[1] Full mapping of Fermat sieves available here: https://www.scribd.com/doc/283873214/Fermat-Sieve-Using-Complex-Numbers

5. Alternative Method

When testing the resulting complex numbers associated with the factors of N, we noticed an alternative computation that could be computed yielding the same result. Unfortunately a visualization to this technique does not exist, but the following example should be clear enough.

5.1 Alternative Example

$N = 8051$

We know that the factors are [83,97] represented by the complex number $(90 + 7i)$. We can focus on the $(7i)$ and reduce the steps from the horizontal Fermat method responsible for identifying this term.

If we average $\left(\lceil\sqrt{\boldsymbol{N}}\rceil + \boldsymbol{N}\right)$ we have $\boldsymbol{N}^*$:

$$\boldsymbol{N}^* = \frac{\left(\lceil\sqrt{\boldsymbol{N}}\rceil + \boldsymbol{N}\right)}{\mathbf{2}}$$

$$\boldsymbol{N}^* = \frac{\left(\lceil\sqrt{\mathbf{8051}}\rceil + \mathbf{8051}\right)}{\mathbf{2}}$$

$$\boldsymbol{N}^* = \frac{(\mathbf{90} + \mathbf{8051})}{\mathbf{2}}$$

$$\boldsymbol{N}^* = 4070.5$$

$$\boldsymbol{b}^* = \frac{\boldsymbol{b}}{\mathbf{2}} = \mathbf{3.5}$$

$\boldsymbol{GCD}[(\boldsymbol{N}^* - \boldsymbol{b}^*), \boldsymbol{N}]$ is the factor of N. $\boldsymbol{GCD}[(\mathbf{4070.5} - \mathbf{3.5}), \mathbf{8051}] = \mathbf{83}$

If we re-iterate this average of $\boldsymbol{N}^*$, $\boldsymbol{b}^*$ is cut in half again.

$$\boldsymbol{N}^{**} = \frac{(\boldsymbol{N}^* + \boldsymbol{N})}{\mathbf{2}}$$

$$\boldsymbol{N}^{**} = \frac{(\mathbf{4070.5} + \mathbf{8051})}{\mathbf{2}}$$

$$\boldsymbol{N}^{**} = \mathbf{6060.75}$$

$$\boldsymbol{b}^{**} = \frac{\boldsymbol{b}^*}{\mathbf{2}} = \mathbf{1.75}$$

$\boldsymbol{GCD}[(\boldsymbol{N}^{**} - \boldsymbol{b}^{**}), \boldsymbol{N}]$ is the factor of N. $\boldsymbol{GCD}[(\mathbf{6060.75} - \mathbf{1.75}), \mathbf{8051}] = \mathbf{83}$

We can re-iterate this average of $\boldsymbol{N}^{**}$, such that $\boldsymbol{b}^{**}$ is cut in half again.

$$\boldsymbol{N}^{***} = \frac{(\boldsymbol{N}^{**} + \boldsymbol{N})}{\boldsymbol{2}}$$

$$\boldsymbol{N}^{***} = \frac{(\boldsymbol{6060.75 + 8051})}{\boldsymbol{2}}$$

$$\boldsymbol{N}^{***} = \boldsymbol{7055.875}$$

$$\boldsymbol{b}^{***} = \frac{\boldsymbol{b}^{**}}{\boldsymbol{2}} = \boldsymbol{0.875}$$

$\boldsymbol{GCD}[(\boldsymbol{N}^{***} - \boldsymbol{b}^{***}), \boldsymbol{N}]$ is the factor of *N*. $\boldsymbol{GCD}[(\boldsymbol{7055.875 - 0.875}), \boldsymbol{8051}] = \boldsymbol{83}$

The initial explanation is rooted in the notion of the product of primes less than $\sqrt{\boldsymbol{N}}$ where

$$F = \prod_{n=1}^{\sqrt{N}} n \;\; where\ F > N$$

the $\boldsymbol{GCD}[\boldsymbol{F}, \boldsymbol{N}] = \boldsymbol{Factor\ of\ N}$. But why these *partial products* $(where\ F < N)$ comprised of factors of *N* are residing near or on iterated averages related directly to the imaginary coefficient of $(a + bi)$ is still a mystery.

5.2 Alternative Example Sieve

There is a very powerful sieve associated with the alternative method based on GCD's. We do not need to check every $\boldsymbol{b}^*$ from every $\boldsymbol{N}^*$ in $\boldsymbol{GCD}[(\boldsymbol{N}^* - \boldsymbol{b}^*), \boldsymbol{N}]$. At the end of every iterated average $\boldsymbol{N}^*$, the remainder will be of a specific number.

- **There is only 1 possible $\boldsymbol{b}^*$ with a matching remainder to generate an integer for $(\boldsymbol{N}^* - \boldsymbol{b}^*)$ each iteration.**
- **Also there is only 1 possible $\boldsymbol{b}^*$ with a complimentary remainder to generate an integer for $(\boldsymbol{N}^* + \boldsymbol{b}^*)$ each iteration.**

Using our example above where $N = 8051$, $\boldsymbol{N^{***}} = \mathbf{7055.875.}$ This is the 3rd iteration of our iterated average. We know from our Fermat sieve that the imaginary coefficient in the $(a + bi)$ has to end in a 3, 5, or 7.

Below are the 3, 5, and 7 series divided by $2^{iteration}$.

$\boldsymbol{b}$	$\dfrac{\boldsymbol{b}}{\mathbf{2}^{\boldsymbol{iteration}}}$	$\boldsymbol{b}$	$\dfrac{\boldsymbol{b}}{\mathbf{2}^{\boldsymbol{iteration}}}$	$\boldsymbol{b}$	$\dfrac{\boldsymbol{b}}{\mathbf{2}^{\boldsymbol{iteration}}}$
3	0.375	5	0.625	7	0.875
13	1.625	15	1.875	17	2.125
23	2.875	25	3.125	27	3.375
33	4.125	35	4.375	37	4.625
43	5.375	45	5.625	47	5.875
53	6.625	55	6.875	57	7.125
63	7.875	65	8.125	67	8.375
73	9.125	75	9.375	77	9.625
83	10.375	85	10.625	87	10.875
93	11.625	95	11.875	97	12.125
103	12.875	105	13.125	107	13.375
113	14.125	115	14.375	117	14.625
123	15.375	125	15.625	127	15.875
133	16.625	135	16.875	137	17.125
143	17.875	145	18.125	147	18.375
153	19.125	155	19.375	157	19.625
163	20.375	165	20.625	167	20.875
173	21.625	175	21.875	177	22.125
183	22.875	185	23.125	187	23.375
193	24.125	195	24.375	197	24.625
203	25.375	205	25.625	207	25.875

Table 1. Required $\boldsymbol{b}$ for $\boldsymbol{N^{***}}$ which must end in .875.

We can see the only possible $\boldsymbol{b}$ to subtract from $\boldsymbol{N^{***}}$ occurs with every value ending in *xx*.875. Thus we do not have to check every integer value of $\boldsymbol{b}$. Table 1 illustrates that the only possible solutions for $\boldsymbol{GCD}[(\boldsymbol{N^{***}} - \boldsymbol{b^{***}}), \boldsymbol{N}]$ exist for

$\boldsymbol{b^{***}} = \{\mathbf{7}, \mathbf{15}, \mathbf{23}, \mathbf{47}, \mathbf{55}, \mathbf{63}, \mathbf{87}, \mathbf{95}, \mathbf{103}, \mathbf{127}, \mathbf{135}, \mathbf{143}, \mathbf{167}, \mathbf{175}, \mathbf{183}, \mathbf{207}, \dots\}$. It should be obvious to note that each possible series is perfectly aligned with a difference of 5.

$$7 \rightarrow 0.875$$

$$47 \rightarrow 5.875$$

This difference of 5 means:

- $(\boldsymbol{N}^{***} - \boldsymbol{b}^{***}) = \mathbf{7053}\,, \mathbf{7048}, \mathbf{7043}$ ... in our routine when testing possible 3's for $\boldsymbol{b}^{***}$.
- $(\boldsymbol{N}^{***} - \boldsymbol{b}^{***}) = \mathbf{7054}\,, \mathbf{7049}, \mathbf{7044}$ ... in our routine when testing possible 5's for $\boldsymbol{b}^{***}$.
- $(\boldsymbol{N}^{***} - \boldsymbol{b}^{***}) = \mathbf{7055}\,, \mathbf{7050}, \mathbf{7045}$ ... in our routine when testing possible 7's for $\boldsymbol{b}^{***}$.

- **It also holds for higher iterations such that the number of possible $\boldsymbol{b}$ per series sieved is equal to $\mathbf{2}^{\boldsymbol{iteration}-\mathbf{1}}$.**

Table 2 below illustrates the 4th iteration for the 3, 5, and 7 series where $\boldsymbol{N}^{****} = \mathbf{7553.4375}$.

$\boldsymbol{b}$	$\frac{\boldsymbol{b}}{\mathbf{2}^{\boldsymbol{iteration}}}$	$\boldsymbol{b}$	$\frac{\boldsymbol{b}}{\mathbf{2}^{\boldsymbol{iteration}}}$	$\boldsymbol{b}$	$\frac{\boldsymbol{b}}{\mathbf{2}^{\boldsymbol{iteration}}}$
3	0.1875	5	0.3125	7	0.4375
13	0.8125	15	0.9375	17	1.0625
23	1.4375	25	1.5625	27	1.6875
33	2.0625	35	2.1875	37	2.3125
43	2.6875	45	2.8125	47	2.9375
53	3.3125	55	3.4375	57	3.5625
63	3.9375	65	4.0625	67	4.1875
73	4.5625	75	4.6875	77	4.8125
83	5.1875	85	5.3125	87	5.4375
93	5.8125	95	5.9375	97	6.0625
103	6.4375	105	6.5625	107	6.6875
113	7.0625	115	7.1875	117	7.3125
123	7.6875	125	7.8125	127	7.9375
133	8.3125	135	8.4375	137	8.5625
143	8.9375	145	9.0625	147	9.1875
153	9.5625	155	9.6875	157	9.8125
163	10.1875	165	10.3125	167	10.4375
173	10.8125	175	10.9375	177	11.0625
183	11.4375	185	11.5625	187	11.6875
193	12.0625	195	12.1875	197	12.3125
203	12.6875	205	12.8125	207	12.9375

Table 2. Required $\boldsymbol{b}$ for N^{****} which is half of the number of required $\boldsymbol{b}$ for $\boldsymbol{N}^{***}$.

Synching the remainders to the iterated average remainder with integer solutions will then allow us to check every 5th integer for $\boldsymbol{b^*}$ in $\boldsymbol{GCD}[(\boldsymbol{N^*} - \boldsymbol{b^*}), \boldsymbol{N}]$. A dramatic savings.

- **<u>Furthermore, this savings is compounded by the fact that every 5th integer captures the number of possible $\boldsymbol{b^*}$ for each increased iteration.</u>**

Our example above shows how we only have to check every 7 counting by 40 for the 3rd iterated average and then every 7 counting by 80 for the 4th iterated average both by using every 5th integer from the iterated average in the GCD.

6. Discussion

We have generated the complex space of possible factors to N. We have also illustrated how each of the familiar methods navigates this space. From these insights we presented a combined method in applying the most efficient method (of those considered) to specific sections of the factor strip while retaining the benefits of sieving.

We have also noted an alternative method to Fermat's horizontal method. Much more research is required on this method. However, for worst case scenario factors for all methods, initial testing reveals the alternative method has noticeably fewer required iterations from a specific iterated average. This alternative method enjoys a parallel processing possibility from each iterated average, since the maximum iterated average length is quite small to N, and is currently estimated at $\log_2\left(\frac{N-9}{6}\right)$.

The alternative method is the beneficiary of a very powerful scalable sieve. As with the explanation of the alignment of factors with the iterated average, much more research is required on the sieve to understand its full capability.

We have provided R Code for all routines in the following section. There are undoubtedly efficiencies to be realized from implementing a more robust language for factorization. We are hopeful that other efficiencies or methods of factorization are realized from this visualization of the complex space.

7. R Code

7.1 Complex Space Generation

```
Complex.Space.Generator <- function(N){

  min.real = ceiling(sqrt(N))

  segtop_x = ceiling(N/6)+1
  segtop_y = floor(N/6)-1

  plot(N,N,xlim=c((0),N/3), ylim =c((0),N/6),xlab="Real",ylab="Imaginary")


  segments(3,0,segtop_x,segtop_y)


  segments(segtop_x,segtop_y,N/3,0, lty = "dotted")


  for (i in 3:N/3){
    for (j in 0:N/3){
      i=as.integer(i)
      j=as.integer(j)

      if((i-j)>=3 && (i+j)<=(N/3) && (i-j)<=min.real)

      points(i,j, pch = 17,col = ifelse((i-j)*(i+j) < N,'blue',ifelse((i-j)*(i+j) == N,'green','red')))

    }
  }
}
```

7.2 Simultaneous Complex Factorization

```
Simultaneous.Complex.Factorization <- function(N){

  min.real = ceiling(sqrt(N))
  max.imaginary = min.real - 3
  divisor = min.real + max.imaginary  ### 135degree yellow line

### Estimate of where complex space turns red
  min.imaginary = floor(mean(c((N/divisor),max.imaginary)))

### Corresponding real to above estimate
  max.real = divisor - min.imaginary

  j = 0L

  while (j>=0L ){

    if(
      ### Vertical Fermat
       sqrt(((min.real + j)^2) - N)%%1==0 |
      ### Horizontal Fermat
       sqrt(((max.imaginary - j)^2) + N)%%1==0 |
      ### Trial Division
       (N/((min.real+j)-(max.imaginary-j)))%%1==0
       |

      ### Vertical Fermat from estimate
       sqrt(((max.real - j)^2) - N)%%1==0 |
      ### Trial Division from estimate
       (N/((max.real-j)-(min.imaginary+j)))%%1==0
       ) {


      return(as.matrix(c(iterations=j,
            mapped.real = (min.real+j)-(max.imaginary - j),
            Factor_v = if((sqrt((min.real+j)^2 - N))%%1==0) (min.real+j)-sqrt((min.real+j)^2 - N),
            Factor_h = if((sqrt((max.imaginary - j)^2 + N))%%1==0) (max.imaginary - j) + sqrt((max.imaginary - j)^2 + N),
            Factor_td = if((N/((min.real+j)-(max.imaginary-j)))%%1==0) (N/((min.real+j)-(max.imaginary-j)))
            ,
            Factor_v.2 = if((sqrt((max.real-j)^2 - N))%%1==0) (max.real-j)-sqrt((max.real-j)^2 - N),
            Factor_td.2 = if((N/((max.real-j)-(min.imaginary+j)))%%1==0) (N/((max.real-j)-(min.imaginary+j)))
            )))}

    j = j + 1L

  }
}
```

7.3 ALTERNATIVE FACTORIZATION

```
Viole.Factorization <- function(N){

  min.real = ceiling(sqrt(N))
  max.imgainary = (N-9)/6


  iterated.average = 0L

  for (i in 2:log2(max.imgainary)){
    iterated.average[1] = mean(c(min.real,N))
    iterated.average[i] = mean(c(iterated.average[i-1],N))
    }

  descending.iterations = floor(iterated.average)
  ascending.iterations = ceiling(iterated.average)

  print(iterated.average)
  j = 0L

  while(j>=0L){
    for(i in 1:length(iterated.average)){
         if( (descending.iterations[i]-j > descending.iterations[i-1] | ascending.iterations[i]+j <
ascending.iterations[i+1]) &&
           GCD(descending.iterations[i]-j,N)>1 && GCD(descending.iterations[i]-j,N)<N
         | GCD(ascending.iterations[i]+j,N)>1 && GCD(ascending.iterations[i]+j,N)<N
         ){

      return(c("Factor"=GCD(descending.iterations[i]-
j,N),"Factor"=GCD(ascending.iterations[i]+j,N),"Iterated.Average.Level"=i,"Iterations"=j))}


      }

    j = j + 1L

  }

}
```

7.4 ALTERNATIVE FACTORIZATION SIEVE STARTING POINT

```
Starting.point.procedure <- function(N,series,iteration){

  min.real = ceiling(sqrt(N))
  max.imaginary = (N-9)/6
  iterated.average = 0L
  descending.iterated.average = 0L

  Starting.point.1 = 0L

  for (i in 2:log2(max.imaginary)){
    iterated.average[1] = mean(c(min.real,N))
    iterated.average[i] = mean(c(iterated.average[i-1],N))
  }

  descending.iterations = floor(iterated.average)
  ascending.iterations = ceiling(iterated.average)

  for (i in 2:log2(max.imaginary)){
    descending.iterated.average[1] = mean(c(min.real,N))
    descending.iterated.average[i] = mean(c(descending.iterated.average[i-1],min.real))
  }

  descending.iterations.2 = floor(descending.iterated.average)
  ascending.iterations.2 = ceiling(descending.iterated.average)

  print(iterated.average)


### Starting point procedure
    while (series >= 1L  && series < max.imaginary){
      if(series/(2^iteration) - floor(series/(2^iteration)) == iterated.average[iteration] -
floor(iterated.average[iteration])){

        return(Starting.point = iterated.average[iteration] - series/(2^iteration))}

      series = series + 10L

      }

}
```