Phase Space Computation

I. THE PARTITION PROBLEM

We do not go over all details here, since they appear on Avishy's draft (attached). In a nutshell, given an array of integers $z_1, \ldots, z_n$ we construct an input function $\psi(x)$ which encodes the problem. Its Fourier transform $\phi(p)$ has the property that $\phi(\pi m)$ is proportional to the number of m-partitions; according to simulations, we believe the modest goal of finding out whether there are any partitions at all is hard enough as it is. To do so, it is sufficient to check whether $\phi(0)$ vanishes or not, thus solving the partition problem. The following sections practical limits.

A. How big can we go?

Let N be the number of entries of the input (and output) function of our Fourier transform. I.e., N in the number of "bins" when performing discretization. We ask the following question: given parameters $n, d_1, \ldots, d_n$ for the partition problem (here we go by Avishy's notations), what is the minimal number of bins N necessary?

Let us denote $d \triangleq \max_k d_k$. We can prove that at least $\Omega(2^d)$ **bins per unit** are required. Intuitively, this should be possible to deduce by examining Eq. 1 in Avishy's draft: the RHS implies that without having precision of at least $2^{-(d-2)}$ in x, one cannot possibly distinguish the function $\psi(x)$ from another one which is computed for a different ("as close as possible") array of inputs $z_1, \ldots, z_n$ (with n fixed). To clarify: denote

$$\psi_{\boldsymbol{z}}(x) = \prod_{k=1}^{n} \cos(\pi z_k x) \tag{1}$$

where $\boldsymbol{z} \in \mathbb{Z}^n$ is an array of n integers. Denote the maximum of this array by z_j. Flipping the most significant bit of the binary representation of z_j yields a slightly different function $\psi_{\boldsymbol{z}'}(x)$:

$$\psi_{\boldsymbol{z}'}(x) = \left[\prod_{k \neq j} \cos(\pi z_k x)\right] \cos\left[\pi\left(z_j - 2^{d-1}\right) x\right]. \tag{2}$$

Assume x has precision δx, and let $x = x_0$ be a value on which both functions agree, i.e. $\psi_{\boldsymbol{z}'}(x_0) = \psi_{\boldsymbol{z}}(x_0)$ (such x_0 always exists - e.g. 0). Then we seek the largest δx s.t. $\psi_{\boldsymbol{z}'}(x_0 + \delta x) \neq \psi_{\boldsymbol{z}}(x_0 + \delta x)$. Let us compute:

$$\psi_{\boldsymbol{z}}(x_0 + \delta x) = \left(\prod_{k \neq j} \cos\left[\pi z_k (x_0 + \delta x)\right]\right) \cos\left[\pi z_j (x_0 + \delta x)\right]. \tag{3}$$

And:

$$\begin{aligned}
\psi_{\boldsymbol{z}'}(x_0 + \delta x) &= \left(\prod_{k \neq j} \cos\left[\pi z_k (x_0 + \delta x)\right]\right) \cos\left[\pi\left(z_j - 2^{d-1}\right)(x_0 + \delta x)\right] = \\
&= \left(\prod_{k \neq j} \cos\left[\pi z_k (x_0 + \delta x)\right]\right) \cos\left[\underbrace{\pi\left(z_j - 2^{d-1}\right) x_0}_{=\pi z_j x_0 + 2\pi m} + \pi\left(z_j - 2^{d-1}\right)\delta x\right] = \\
&= \left(\prod_{k \neq j} \cos\left[\pi z_k (x_0 + \delta x)\right]\right) \cos\left[\pi z_j (x_0 + \delta x) - 2\pi \delta x \cdot 2^{d-2}\right]
\end{aligned} \tag{4}$$

If x only has precision $\delta x = 2^{-(d-2)}$, the two functions are indistinguishable for $x_0 + \delta x$ as well. Thus, we require at least $\delta x \geq 2^{-(d-1)}$ - i.e., the required precision is about twice the reciprocal of the largest input integer.

Now let discuss the range of x. By Avishy's notations, the range is $[-\alpha, \alpha]$; and by his analysis, α should be in the order of n. Thus, we conclude that $N = \Omega\left(n \times 2^d\right)$.

There is one more thing we would like to point out: if one performs a similar analysis by observing Avishy's Eq. 1 instead, one could suggest that the required precision is proportional to the reciprocal of $\sum_{k=1}^{n} z_k$ - i.e., the *sum* of all integers, and not just their maximum. However, we are not entirely certain this argument is valid.

B. Bottom Lines - The Experiment

- Felix, with his equipment, can generate an input function (for the Fourier transform) with $N \approx 1000$ bins. Based on the analysis above, each integer should satisfy $n \times 2^{d-1} \lessapprox N$. Thus, we recommend to perform the experiment with input integer arrays of size $n = 10$, with each integer no greater than roughly 100.
- Felix said that it is simpler to perform only a phase modulation (without altering the amplitude), which would compute a Fourier transform for a function of the form: $\psi(x) = N \exp\left[-\frac{x^2}{2\sigma^2} + if(x)\right]$ where $f(x)$ is a real function - i.e., a Gaussian where we alter the complex phase for each x as we wish. Felix suggested that if we could somehow come up with an idea for how to solve the partition problem with this kind of input function, it would have been very interesting (and also, he might be able to achieve better resolution). However, it seems this would require a very complicated trick - something in the spirit of Shor's algorithm.

C. Simulations

We have performed simulations for two integer arrays:

$$\boldsymbol{z}_1 = \{43, 91, 18, 26, 15, 14, 87, 58, 55, 14\},$$
$$\boldsymbol{z}_2 = \{85, 62, 35, 51, 40, 8, 24, 12, 19, 24\}.$$

The first array has no partition, however it has the following 1-partition:

$$43 + 91 + 18 + 58 = 210, \qquad 26 + 15 + 14 + 87 + 55 + 14 = 211 \tag{5}$$

The second one does have a partition:

$$85 + 35 + 40 + 8 + 12 = 62 + 51 + 24 + 19 + 24 = 180. \tag{6}$$

For each array we have performed two simulations: one with high resolution (which we cannot achieve in experiments), and one conforming to the constraints we have detailed in the previous sections. For each simulation we plot both input and output of the Fourier transform.

In the output functions, the "time" axis has units of 2π; thus, the value for $p = 1/2$, for example, should be proportional to the number of 1-partitions; as we mentioned before, this does not seem to work very well.

The high-precision simulation for $\boldsymbol{z}_1$, plotted in Fig. 1,2 has very clean output.

The low-precision simulation for $\boldsymbol{z}_1$ (Fig. 3,4) has less clean of an output, however one may clearly see that it vanishes at $p = 0$ - i.e., no partition. The high-precision simulation for $\boldsymbol{z}_2$, plotted in Fig. 5,6, has very clean output. The low-precision simulation for $\boldsymbol{z}_2$ (Fig. 7,8) also has nice output. One may clearly see that it has a maximum at $p = 0$ - i.e., there is a partition.

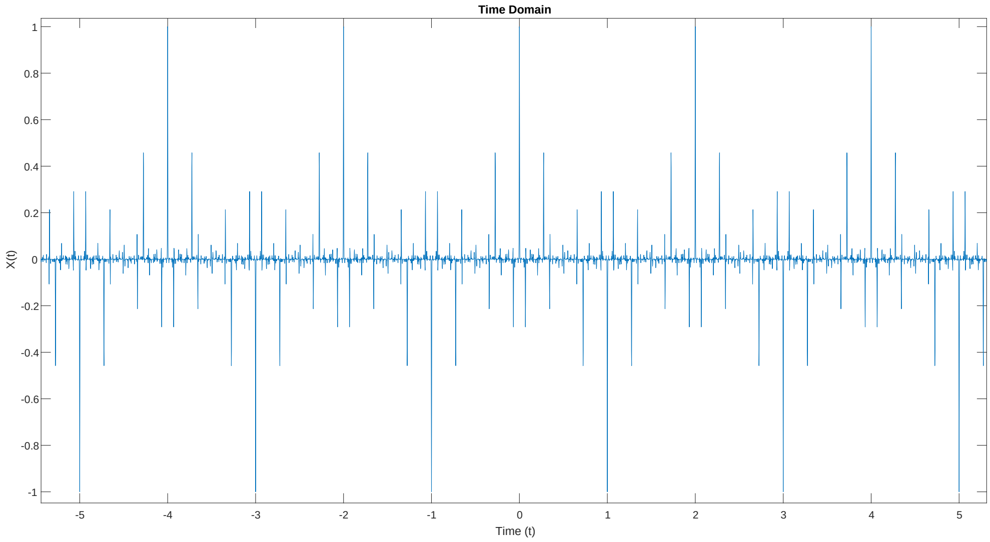

FIG. 1: Plot of the input function $\psi(x)$ for array $\boldsymbol{z}_1$, with range $\alpha = 50$ and precision $\delta x = 1/500$.

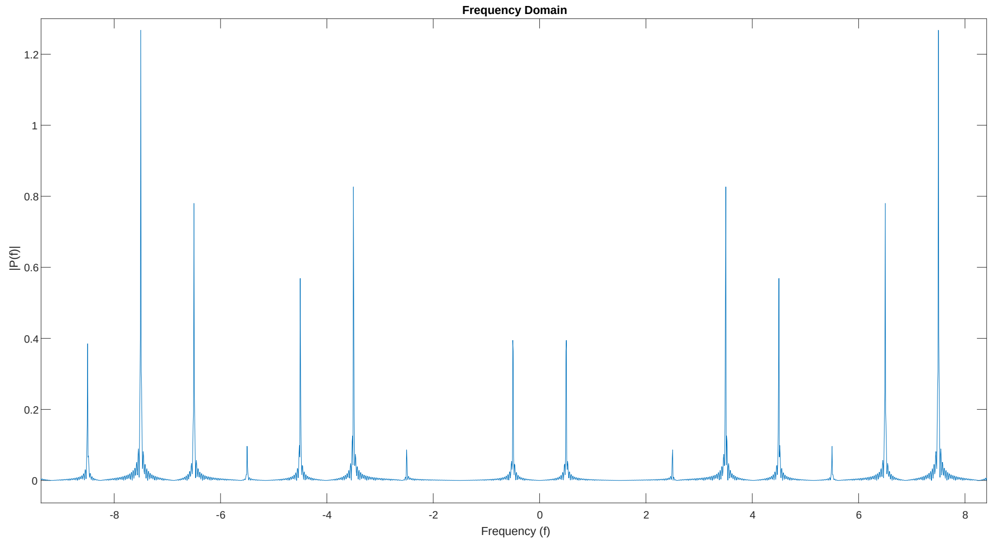

FIG. 2: Plot of the output function $\phi(p)$ for array $\boldsymbol{z}_1$, with range $\alpha = 50$ and precision $\delta x = 1/500$.

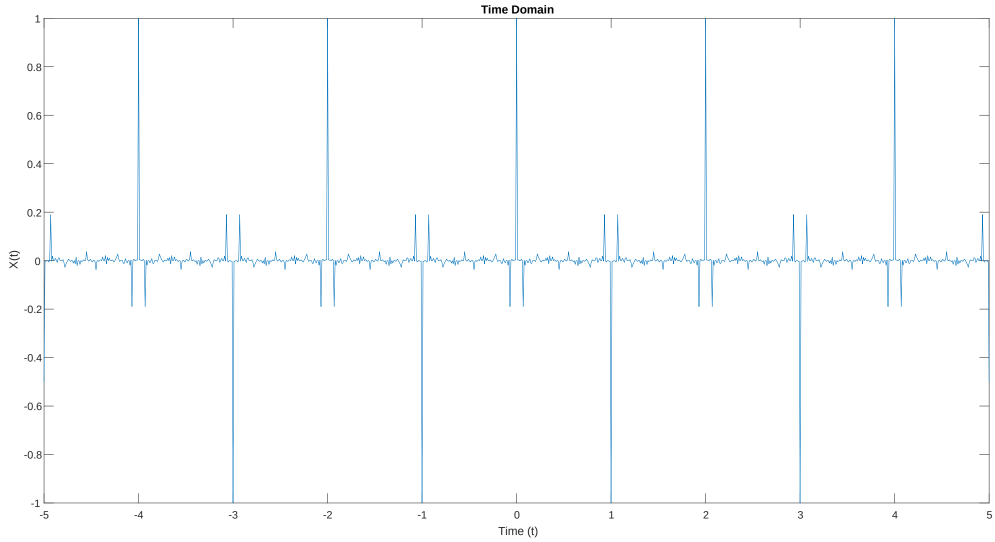

FIG. 3: Plot of the input function $\psi(x)$ for array $\boldsymbol{z}_1$, with range $\alpha = 5$ and precision $\delta x = 1/100$.

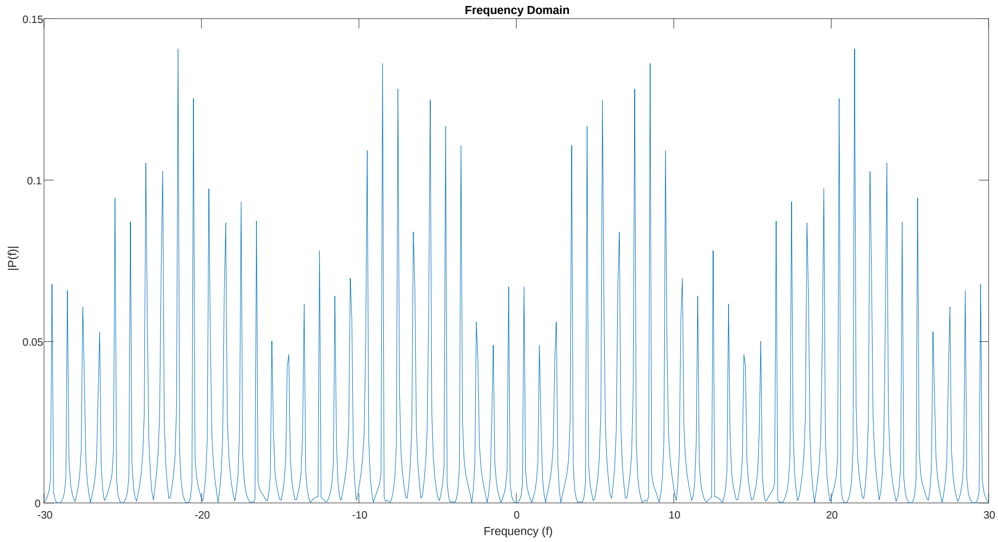

FIG. 4: Plot of the output function $\phi(p)$ for array $\boldsymbol{z}_1$, with range $\alpha = 5$ and precision $\delta x = 1/100$.

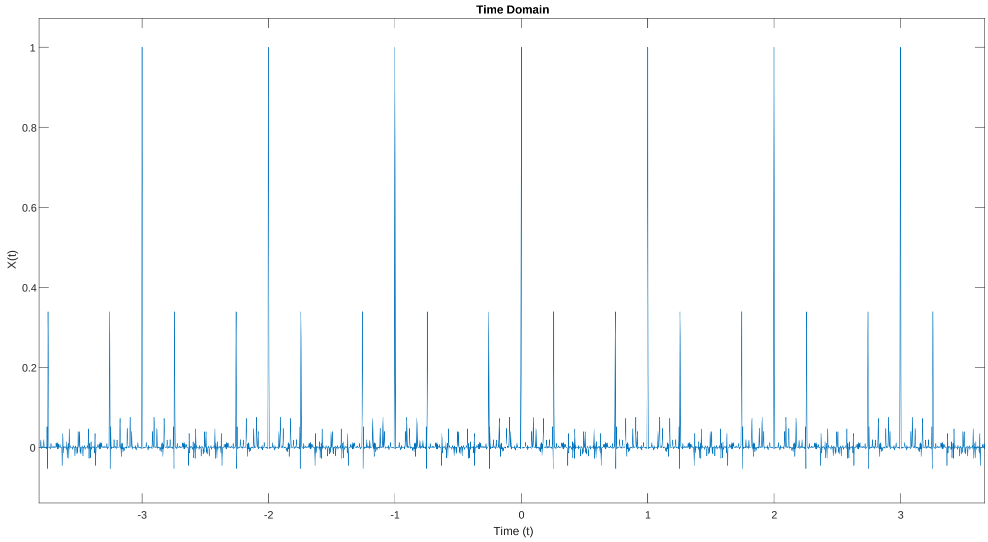

FIG. 5: Plot of the input function $\psi\,(x)$ for array $\boldsymbol{z}_2$, with range $\alpha = 10$ and precision $\delta x = 1/500$.

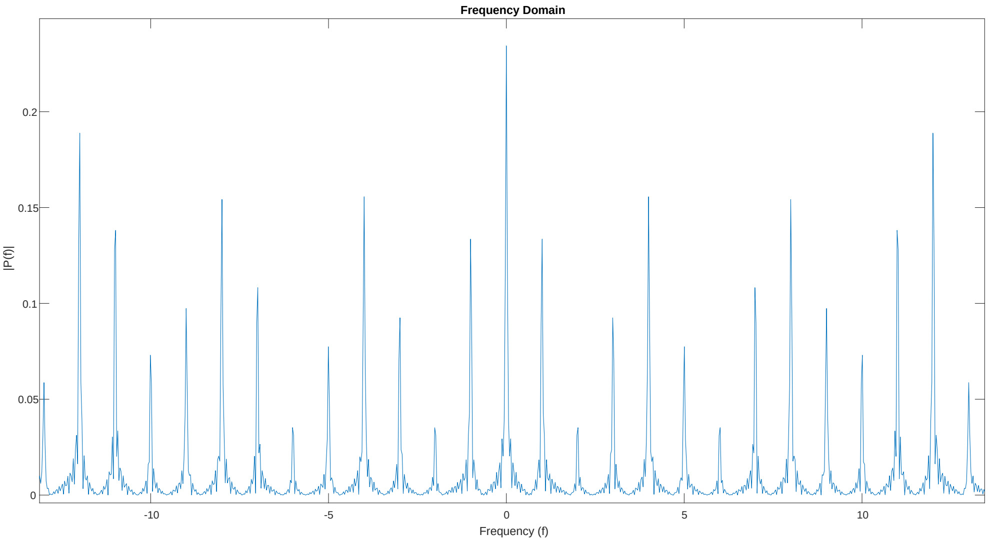

FIG. 6: Plot of the output function $\phi\,(p)$ for array $\boldsymbol{z}_2$, with range $\alpha = 10$ and precision $\delta x = 1/500$.

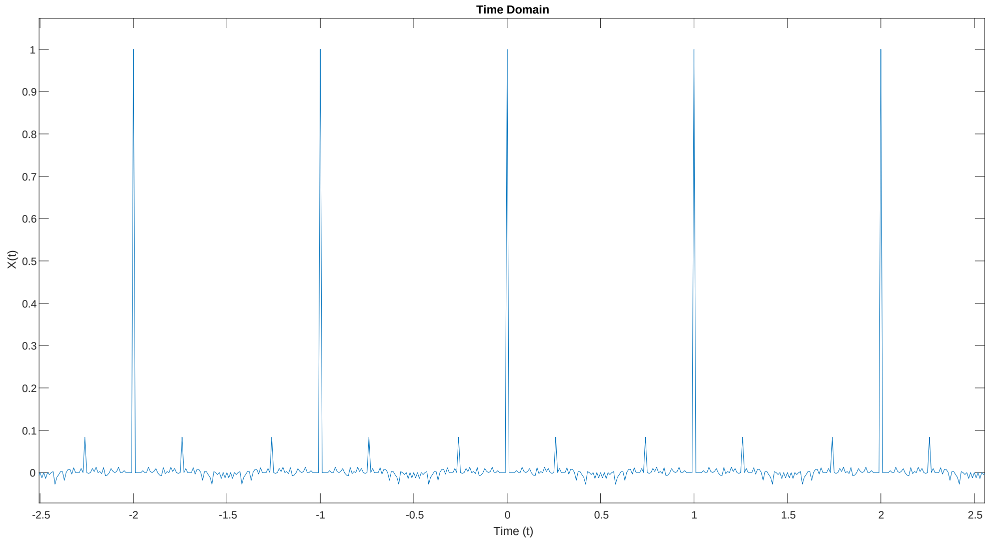

FIG. 7: Plot of the input function $\psi(x)$ for array $\boldsymbol{z}_2$, with range $\alpha = 5$ and precision $\delta x = 1/100$.

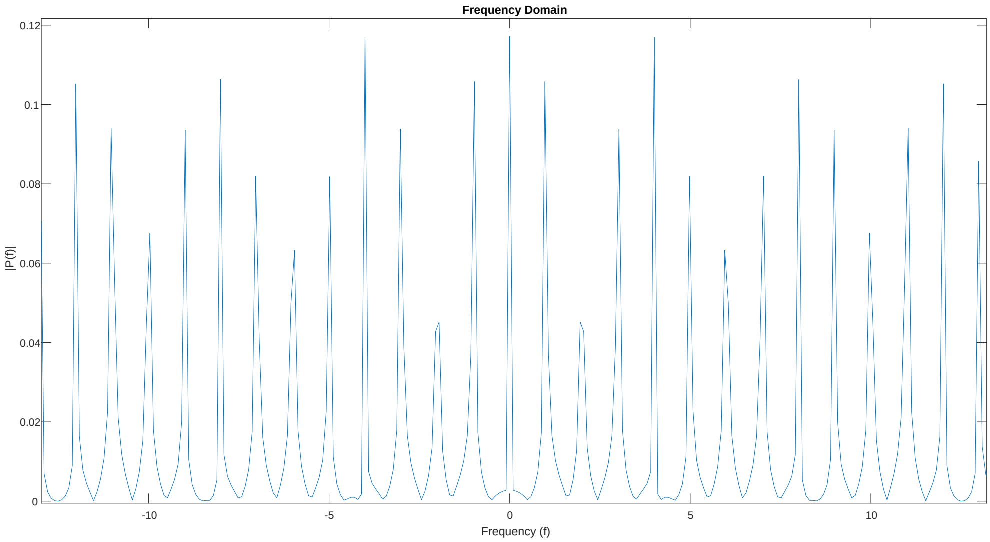

FIG. 8: Plot of the output function $\phi(p)$ for array $\boldsymbol{z}_2$, with range $\alpha = 5$ and precision $\delta x = 1/100$.

II. THE ZEROS OF THE RIEMANN ZETA FUNCTION

Sir Michael Berry has authored two drafts (attached), proposing several appropriate "input functions" $F\left(\frac{x}{x_0}\right)$ - i.e., functions (x_0 is a scaling factor). These all have the property that their Fourier transform $G(t)$ has exactly the same zeros as $\zeta\left(\frac{1}{2}+it\right)$ for real t. In other words, $G(t_0)=0$ iff t_0 is the imaginary part of a nontrivial zero of the Riemann zeta function, located on the critical line.

According to Berry's second paper, the one with a Gaussian modulation factor is the most promising candidate for experiment; thus, we focused in studying it and attempting to repeat his work. This is the input function $F(u)$ appearing in Eq. B.3.4 (hereon, equation and figure numbers starting with a 'B' refer to Berry's second paper). There seem to be at least one or two errors in this equation; we tried our best to find and fix any errors, but even after these corrections we were unable to reproduce a function resembling Fig. B.2.f. Our closest attempt appears in Fig. 9. The next thing we did was aiming straight for the desired result. We took the Fourier transform's required *outcome*, $G(t)$ (B.1.4 combined with B.3.1), and then computed its *inverse* Fourier transform. As you may see in Fig. 11, this time we achieved a better match.

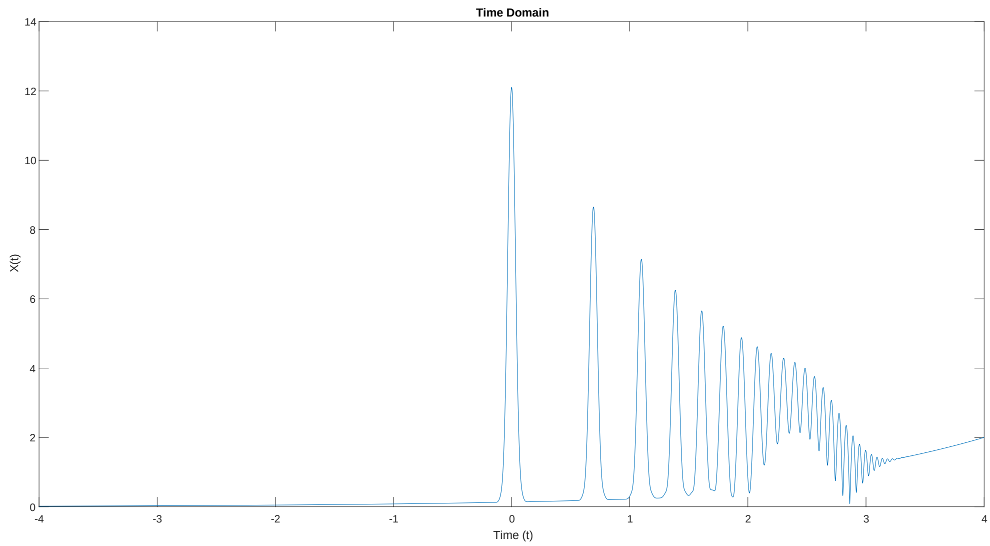

FIG. 9: Plot of $|F(u)|$ taken from (the corrected form of) Eq. B.3.4

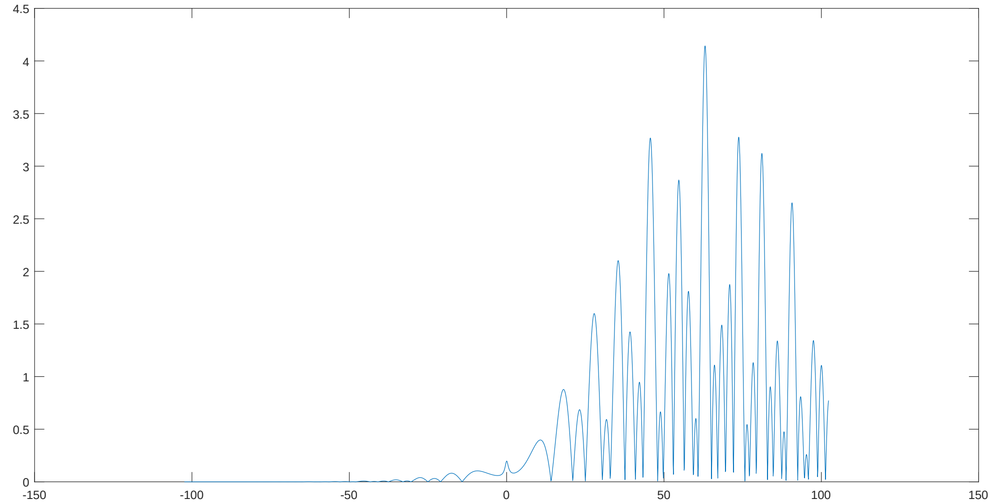

FIG. 10: Plot of $G\left(t\right)$, having zeros in $t = 14, 21, 25, 30.4, 33$ etc. - precisely where $\zeta\left(\frac{1}{2} + it\right)$ vanishes.

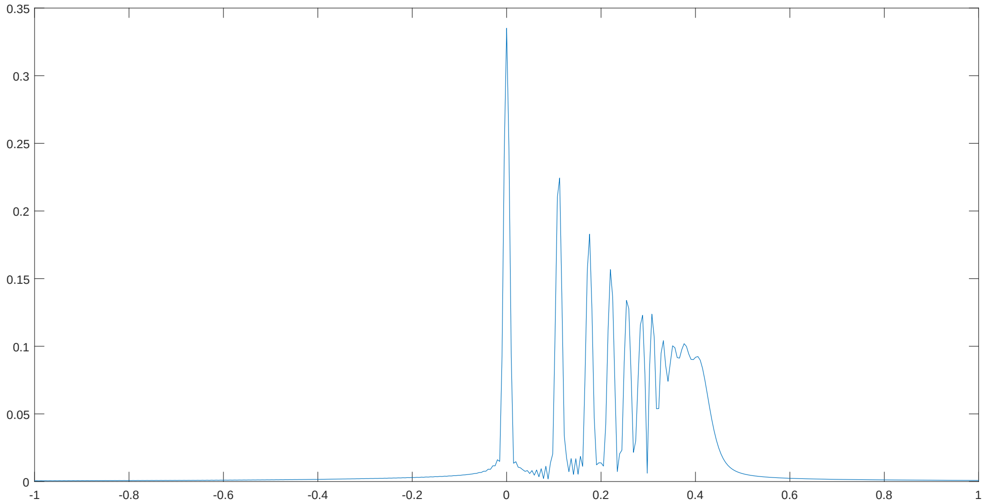

FIG. 11: Plot of $F\left(u\right)$, taken to be the inverse Fourier transform of $G\left(t\right)$. Compare with Fig. B.2.f.