XM project
Equidistribution of sequences related to Ulam's numbers

Alexei Condurachi, Jacob Bast Hall, Malte Kjellerup Juhl

January 2021

Contents

1 Introduction

The main goal for this project is to study the behaviour of the Ulam sequence. First of all one should present how it is defined: given two initial values a_1 and a_2, a_n is the smallest integer bigger than a_{n-1} that can be written **uniquely** as a sum of two **different** previous numbers. The sequence is present in the OEIS database (A2858).

For instance, if we consider the standard initial values $a_1 = 1$ and $a_2 = 2$ the sequence is

$$1, 2, 3, 4, 6, 8, 11, 13, 16, 18, 26, 28, 36, 38, 47, 48, 53, 57, 62, 69, \ldots \tag{1}$$

If we then consider as initial values $a_1 = 1$ and $a_1 = 3$ we obtain

$$1, 3, 4, 5, 6, 8, 10, 12, 17, 21, 23, 28, 32, 34, 39, 43, 48, 52, 54, 59, \ldots$$

We will focus on (1).

A study of the partial Fourier series

$$f_N(x) = \Re \sum_{n=1}^{N} e^{ia_n x} = \sum_{n=1}^{N} \cos(a_n x) \tag{2}$$

where the coefficients a_n are the Ulam numbers, reveals that there is a special frequency $x = \alpha \sim 2.5714474985$, where the series seems to presents a linear growth in N, which was not expected.

This would mean that the dilated sequence αa_n has non-uniform behaviour mod 2π. In the project we try and search for other values β such that βa_n behaves non-uniformly mod 2π. The standing hypothesis in the literature is that this interesting behaviour occurs for $\beta \in \{k\alpha \mid k \in \mathbb{N}_0\}$ and no other other values.

Furthermore, it looks like the phenomenon is the same for any "non-degenerate" initial values (see the Theory section for more on this); i.e. there always exists such an α depending on the initial conditions.

2 Theory

For any integer sequence, it is natural to ask if it is equidistributed:

Definition 2.1. We say that a sequence $(x_n)_{n\geq 1} \in \mathbb{R}$ is equidistributed modulo 1 if

$$\lim_{n\to\infty} \frac{\#\{n \mid \langle x_n \rangle \in [\alpha, \beta]\}}{n} = \beta - \alpha$$

where $0 \leq \alpha < \beta \leq 1$ and $\langle x \rangle$ is the fractional part of x. Weyl's Criterion then states that

$$(x_n)_n \text{ equidistributed modulo } 1 \iff \sum_{n\leq N} e^{2\pi k x_n} = o(N), \quad \forall\, k \in \mathbb{Z} \setminus \{0\}$$

Inserting $k = \pm 1$ then gives

$$\sum_{n\leq N} \frac{\left(e^{2\pi i x_n} + e^{-2\pi i x_n}\right)}{2} = \sum_{n\leq N} \cos(x_n) = o(N)$$

In fact for general frequencies, we expect $f_N(x) = O(\sqrt{N})$ (this is like a random walk). But our takeaway is that if ever $\sum_{n \leq N} \cos(\beta a_n) = \Theta(N)$ for some β, then βa_n is not equidistributed. This gives us a strategy for searching for interesting β's, namely by looking for linear growth of the cosine sum.

There is another theorem stated on the Wikipedia page for Equidistributed sequence (they give a reference to Felix Bernstein, 1911):

Theorem 2.1. *For any sequence of distinct integers $(a_n)_n$, the sequence $(\beta a_n)_n$ is equidistributed mod 1 for almost all values of β.*

In other words, the set of interesting β's have measure 0. The hypothesis is that they are given exactly by $\beta = k\alpha$ for $k \in \mathbb{N}_0$. Note that we should not be surprised that $k\alpha$ works (it is almost like realizing that twice the even numbers are still even); the interesting part is that these are all of them. Note that this also means that we can use all the $(k\alpha a_n)$-sequences (using the different k, that is) to tweak the value of α.

What happens if the initial conditions (a_1, a_2) are different? According to the literature, two things may happen: There is a degenerate case (not an generally used term), where the consecutive differences eventually become periodic. There are various results on when this is guaranteed to happen, one of which is with $(a_1, a_2) = (2, n)$ for any odd $n \geq 5$ (Schmerl and Spiegel, 1994). But in the non-degenerate case (sometimes called "chaotic" in the literature), it seems like the same phenomenon occurs, where one can find some frequency α, dependent on the particular sequence, that works in the same way as described above.

3 Data Generation

We generate the first $(10^7 - 1)$ of the standard Ulam numbers using the linear time algorithm discussed below. In addition, we use our own quadratic time algorithm to generate Ulam-type sequences up to 10^6 (note that this is a cap on the values, not the number of values) for initial values of $(1, 3)$, $(1, 4)$, $(1, 5)$, $(2, 3)$, $(2, 4)$ and $(3, 4)$.

It took about $5 \sim 6$ minutes to generate each of these sequences using our own algorithm, and due to the quadratic complexity the computing time quickly blows up for higher values. We don't really need more data for our current purposes anyway.

3.1 Our Algorithm

We have implemented in Python the following fairly naive algorithm for generating the Ulam sequences running in $O(M^2)$ time, where we find all Ulam numbers up to M. The algorithm uses a sieve method very similar to the sieve of Eratosthenes. At a conceptual level, it works like this:

- We compute the numbers sequence iteratively. Let L be the current list of Ulam numbers.
- Let the sieve S contain all integers that are a unique sum of two different numbers in L, but excluding the numbers already in L. In this way, S contains candidates for coming Ulam numbers.
- **Repeat:** The next Ulam number x is the minimum number in S. Update S by deleting x from S and considering all sums in $x + a$ for $a \in L$. Add x to L.

Note that we "consider" each sum of two Ulam numbers exactly once with this method, which is the benefit of using a sieve.

To implement the above in practice, we need to keep track of whether a candidate number has so far been a sum 0, 1 or more than 1 times (let's call this the multiplicity). The numbers with a multiplicity of 1 correspond to S above.

In our implementation, we store the numbers with a multiplicity of 0 or 1 as keys in a sorted dictionary, and with the multiplicity as the value. Whenever the multiplicity of a number surpasses 1, we delete it from the dictionary. The next Ulam number is the lowest key in the dictionary with a multiplicity of 1. If x is the current Ulam number we are adding, then $x + L \subset [x, 2x)$, so the dictionary only contains keys in this range at any time (otherwise there would be infinitely many keys with a value of 0).

Time complexity: Even though the sieve method is faster than naively checking all sums of L each time we want to test if a number should be added to the sequence, we unfortunately still get quadratic time. We have $|L| = O(x)$ at any time, so the number of updates to the dictionary is $O(x)$, and each update takes $O(1)$ time on average. So the full complexity is

$$\sum_{x=1}^{M} O(x) \cdot \mathbb{1}(x \text{ is Ulam number}) \leq \sum_{x=1}^{M} O(x) = O(M^2)$$

(Note that if the Ulam have positive natural density, then $O(M^2) = O(N^2)$, where N is the number of terms we find of the sequence. But if the Ulam numbers have a natural density of zero, the complexity may actually be $o(N^2)$. The literature does not seem to agree on this point at the moment, but it's not important for our purposes.)

Performance: With our algorithm run on a standard laptop, it takes about 9 seconds to generate the Ulam sequence up to $2 \cdot 10^5$, and about $5 \sim 6$ minutes to generate the sequence up the 10^6. We might run into memory problems if we were to go much further (say if we ran it for several hours), because the dictionary and output list are both of size $\Theta(N)$.

3.2 A Faster Algorithm

A faster way to compute the Ulam numbers is discussed in one of cited documentation. Now we will briefly discuss it and explain the underlying idea of it, since it is well explained in [4].

The algorithm presented, which has been implemented in Java, uses the results found for $\alpha \sim 2.5714474985$. This is translated to the fact that, when we consider the Ulam numbers modulo $\lambda = \dfrac{2\pi}{\alpha}$, the residues *mod* λ lie in the central half of the range for all but four Ulam numbers. Using this result and sorting them according to their residue *mod* λ, the maximum number of steps needed to determine if an integer is either in the sequence or not is reduced to 5. Because of this the efficiency heavily depends on the time needed to keep the list of residues sorted.

To express how efficient this algorithm is, it suffices to say that even on ordinary computers it is expected to take less than an hour to compute up to one billion Ulam numbers. Due to memory space limitation, it was possible to compute (in less than a minute) just the first ten millions elements of the sequence.

Furthermore we notice that the time needed to compute the sequence grows linearly.

4 Experimental Results

4.1 Investigation of the Fourier sums

In our investigations, we use the most precise estimate of α that we have found, which is given in [3]:

$$\alpha \approx \frac{2\pi}{2.443442967784743433} \approx 2.571447498263977693$$

Figure 1 shows $f_N(x) = \sum_{n=1}^{N} \cos(a_n x)$ for $N \in \{100, 1000\}$. Note that by the symmetries of cosine, $f_N(x) = f_N(2m\pi \pm x)$ for any $m \in \mathbb{Z}$, so we only plot the range $[0, \pi]$. The spike at $x = 0$ is trivial of course, but we already see a clear spike at $x = \alpha$ in f_{100}. In f_{1000} more spikes emerge. Let's write $\langle x \rangle$ for the unique value in $[0, \pi]$ such that $\langle x \rangle = 2m\pi \pm x$ for some $m \in \mathbb{Z}$ (i.e. we take mod 2π and then possibly reflect round π). We add lines at $\langle k\alpha \rangle$ to the plots for $k \in \{0, \ldots, 10\}$, and we see that they line up beautifully with the spikes.

Now let's look more closely at how $f_N(x_0)$ grows with N for fixed values of x_0. Look at Figure 2. The plot with $x = 1$ is typical for most values of x (1 was chosen arbitrarily). In contrast, at $x = \alpha$ we see a striking linear growth. One of the main hypotheses is that a linear growth applies for all $x = k\alpha$. We see in the figure that it holds at least all the way up to $k = 50$, which is testament to how finely the constant α is already tuned. At $k = 100$, the linearity is gone however.

In the three plots in the middle line of figure, we try to perturb α slightly with small ϵ to see how sensitive the value is. $\epsilon = 10^{-7}$ is enough to clearly break linearity at $x = \alpha$, but $\epsilon = 10^{-9}$ is not. This is likely related to the fact that we have around 10^7 terms of the sequence. However, for $k = 7$ the linearity already breaks already with $\epsilon = 10^{-9}$; we mention this as a possible way to tweak the value of α.

4.2 Distributions

Recalling the results discussed so far, we can introduce the set $S_N := \{\alpha a_n - 2\pi \lfloor \frac{\alpha 2\pi}{2\pi} \rfloor \mid 1 \leq n \leq N\}$. We expect that S_N has a uniform distribution if our sequence is equidistributed. We can see in in Figure 5 that this is not the case for the value of α we have previously presented. This enables us to study the behaviour of the sequence for different values of α and N.

First we can verify if the distribution of S_N is still not uniform when we use multiples of α instead of α itself. We can see in Figure 5 (where 6α has been chosen only for visual purposes) that our claim holds true.

Moreover, studying how the distribution changes with regards to the α it is given allows us to estimate the value of α (we have to recall that we are using an approximation that has been extrapolated from the Fourier sums, in correspondence with its spikes). From Figure 6 it is easy to see that we need at least 7 significant digits in order to have a distribution that is visibly not uniform (which confirms what we have stated in the previous paragraph). In addition we can also set boundaries for the value of α. In fact from the experiments we see that $2.5714474 \leq \alpha \leq 2.5714476$ since for those extreme points the corresponding distributions tend to be almost uniform (since we are working with a limited set of Ulam numbers this is not clearly visible at first glance, increasing the data set would make it more clear).

4.3 Pixel plot

One tool that can be very useful when we try look for patterns in a sequence of integers is to represent the sequence as a boolean matrix in correspondence with our entries. After several attempts and iteration it seems that using matrices with 108 columns makes the distribution of Ulam numbers quite deterministic. We see a sort of linear disposition for them. If we study just the first 108×108 numbers it would appear that the Ulam's numbers are disposed mainly in vertical lines. Zooming out it we clearly see that this is not the case, as we can observe in Figure 3, that can be found in the Appendix.

However, changing the number of columns (which corresponds to changing the modulo we are considering) we can see that they tend to lie on parallel lines (Figure 4 in the Appendix). It is clear then that the first intuition, that the modulo 108 was a good finding, appears to be incorrect. This can be explained with the fact that we considered just a portion of our data set.

We then prove that there seems not to exist n such that the pixel plot presents vertical lines, i.e. that the Ulam numbers are equidistributed modulo n $\forall n \in \mathbb{N}$. This has been done by studying the variance of how many of them are equal to $k \ mod \ n$. It appears, experimentally, that the variance decreases when n increases, therefore the largest variance is obtain for n equal to 3 (the computation was made with $n \geq 3$; using the condition $n \geq 2$ we obtain that the largest variance occurs for $n = 2$).

In fact, in [3] the authors argue that the clusters we see have a period of

$$\frac{2\pi}{5\alpha - 4\pi} \approx 21.602\ldots$$

which supports our result that no integer n "works".

4.4 Counterexamples

In this section we used pseudorandom code to try to find values x so that

$$N \mapsto f_N(x) = \sum_{i=1}^{N} \cos(a_i x)$$

grows linearly (equivalently $\frac{1}{N}\sum_{i=1}^{N} \cos(a_i x)$ converges to a non-zero constant for $N \to \infty$). This behaviour seems to only be exhibited by values of the form $k\alpha$ modulo 2π and we calculated $\frac{1}{N} f_N(k\alpha)$ for $1 \leq k \leq 20$ and $N = 10,000,100,000,1,000,000$ in Table 1 in the appendix down below. Note that by symmetry $\cos(-x) = \cos(x)$ we can find a unique $a_k \in [0, \pi]$ so that $\frac{1}{N} f_N(k\alpha) = \frac{1}{N} f_N(a_k)$, this is the value in the parentheses in Table 1 below (note that by mistake the alpha used here is not the best estimate of α, but is very close).

In the search of other values x exhibiting linear growth of $N \mapsto f_N(x)$, we used a for loop where we generated a uniformly chosen number, a, in the interval $[0.001, 3.141]$ with 10 digits. As by periodicity and evenness of cosine the function $x \mapsto f_N(x)$ on the interval $[0, \pi]$ has the same image as on the entire real line. We didn't take on the full interval $[0, \pi]$, so that it didn't pick values very close to 0 or π as these has a very high mean. Then we defined the function

$$x \mapsto \cos(ax)$$

and applied it to a list of the first 10,000 Ulam numbers. We then included the entry $[a, \text{Mean of Ulam list}]$ in another list. This was done 10,000 times and took between 1 and a half hour and 2 hours in maple. This was done 10 times (so 100,000 data points in all) and each time all values with mean higher than 0.030 (this was an arbitrarily chosen bound) was written down in Table 2 in the appendix down below (ranked by mean value). In addition it was noted if it was close to any $k\alpha$ for $1 \leq k \leq 20$ (we didn't specify how close it needed to be as we didn't have a sense of what would be a good maximal difference, however all the numbers below are equal to their approximates up to 4 digits when rounded up). Since the function

$$a \mapsto \frac{1}{10,000} f_{10,000}(a)$$

is obviously continuous, values close enough to $k\alpha$ should also have a high mean. The results are in Table 2 in the appendix down below (The table is split into 2 as it was very long).

Two values were found that wasn't close to a $k\alpha$, in these cases, we ran the same code as above, but with a few modifications. We used 100,000 Ulam numbers instead of 10,000 and we generated 500 (instead of 10,000) random numbers in an area around that number (instead of in [0.001,3.141]). Then we found the value with the maximum mean and noted it. In our 2 cases we had:

- **2.680627864**: The maximum mean value in the interval [2.6795,2.6815] found was 2.680570462 with mean -0.00867.
- **2.171397916**: The maximum mean value in the interval [2.1704,2.1724] found was 2.170727355 with mean 0.00783.

We did the same experiment with sine (8 instead of 10 experiments because of time issues, so 80,000 data points in all) and got the following results in Table 3 in the appendix down below (the table is split into 2 as it was very long):

Here 4 values were found that wasn't close to $k\alpha$ for $1 \leq k \leq 20$. We did the same for these values as above (interestingly 2 of the values were very close so we just used the same experiment for both):

- **0.2661285736**: The maximum mean value in the interval [0.2651,0.2671] found was 0.2661335994 with mean 0.01173.
- **2.893332718 and 2.893329854**: The maximum mean value in the interval [2.8923,2.8943] found was 2.893318409 with mean -0.00931.
- **3.022839268**: The maximum mean value in the interval [3.0218,3.0238] found was 3.022635522 with mean -0.01135.

In both experiments most of the values were close to $k\alpha$, which supports our hypothesis. The few values for which this was not the case, the mean was below 0.037 and all the 500 values found around the values, using 100,000 Ulam numbers, had a much lower mean than originally. A similar behavior can however also be seen for example when calculating 10,000 mean and 100,000 mean of 15α, which is also a bit of a decrease, so this doesn't have to mean anything. Obviously the values of $k\alpha$ with mean less than 0.030 at 10,000 Ulam numbers were never detected in the cosine experiment, except for 11α and 14α, as some of the values close to these (and values close to some other $k\alpha$) gave us higher values than in the table above, so either these are not very close to the

"true value" of $k\alpha$, or for a low number of Ulam numbers some values around the "true value" of $k\alpha$ has a higher mean. Curiously 2α was only detected once in the cosine experiment and only with a mean of -0.033, despite having an actual mean of 0.289, and 8α was never detected despite having a mean of 0.105. This means that this method is clearly flawed as it didn't detect values with high mean that we know exists, so we could have missed others.

5 Conclusions

Our hypothesis that $N \mapsto f_N(x) = \sum_{n=1}^{N} \cos(a_n x)$ being linear implies that $x = k\alpha$ (up to symmetries) for some $k \in \mathbb{N}_0$, is supported by our findings so far. Calculations of $f_N(k\alpha)/N$ for high N and graphical representations of $f_N(\alpha)$ indicate that $N \mapsto f_N(k\alpha)$ is linear, at least for k small enough. The spikes from the graphs of our plots of $f_{100}(x)$ and $f_{1000}(x)$ seems to align nicely with $k\alpha$ modulo 2π, indicating that these are the only ones, though this is a purely visual argument and were done for low N.

The pseudorandom experiments only found counterexamples with a relatively low mean value, that seems to decrease further when increasing N, and therefore can possibly be written off as anomalies for low N. The pseudorandom experiments were however not able to detect 8α and only just detected 2α, despite both having a high mean, so this experiment is clearly flawed and will most likely need more data points to be more reliable.

6 Future Work

- A major thing we wanted to look at was other initial conditions on the Ulam numbers, and examine if they exhibit similar behavior. We have the data at hand already, but did not get around to investigate it. In particular examining for which initial values a value similar to alpha exists, as other theoretical work we've read suggests that in some cases such a value seem to exist and sometimes it doesn't as discussed in the theory chapter. So looking at the pairs of initial values on the Ulam numbers which exhibit such behaviour could be of interest.

  Furthermore, it would be interesting to determine the corresponding α-values. In [2], a few are reported:

  $$\begin{aligned} \alpha_{(1,3)} &\sim 2.83349751\ldots \\ \alpha_{(1,4)} &\sim 0.506013502\ldots \\ \alpha_{(2,3)} &\sim 1.1650128748\ldots \end{aligned}$$

  One could verify these and try find the α for other initial conditions.

  Lastly it could be interesting to compare the distributions S_N for different initial condition. Do the distributions share similar qualitative traits? To do this, we first need to know the corresponding α-values.

  We upload data files with Ulam-type sequences for various initial conditions, as well as distribution plots for the initial conditions with known α's listed above.

- Another thing we want to look at would be to make the pseudorandom code more time efficient, as it clearly seems that more data points are needed when searching for values with

a high mean is needed to make it reliant, as the experiment wasn't able to detect some values of interest that we knew existed. Doing the experiments for more Ulam numbers could be interesting, however it could be argued that values with high mean would be even harder to detect since the points around values with high mean should converge to 0. Meaning that we need to to detect a value with high mean even more accurately than we would with only 10,000 Ulam numbers and therefore even more data points would be needed to compensate for this making the runtime very long.

If the code can be made sufficiently efficient (perhaps using Python?), we might be able to even do a thorough search in small increments of x and stop using random samples.

7 Appendix (Figures and Tables)

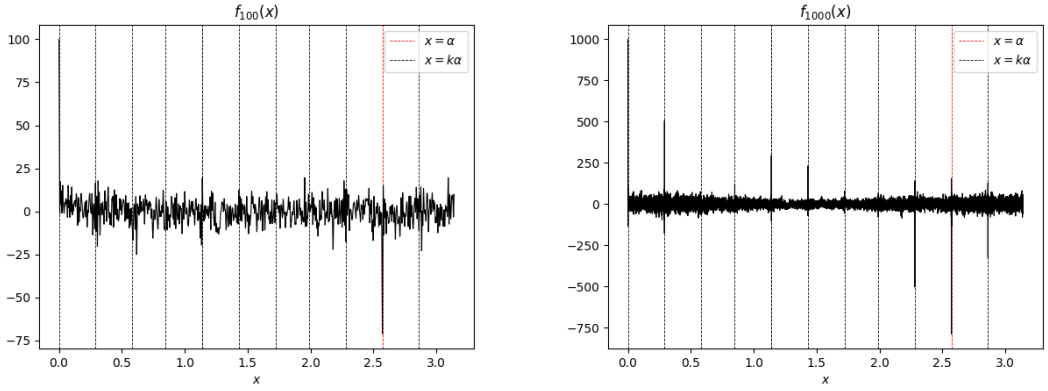

Figure 1: $f_N(x) = \sum_{n=1}^{N} \cos(a_n x)$ for $x \in [0, \pi]$ and $N \in \{100, 1000\}$. The vertical lines are $x = \langle k\alpha \rangle$ for $k = 0, \ldots, 10$.

Table 1: The values of $N^{-1} \sum_{n=1}^{N} \cos(k\alpha a_n)$ for $1 \leq k \leq 20$ and $N = 10^4, 10^5$ and 10^6 respectively.

Value	10,000 Ulam	100,000 Ulam	1,000,000 Ulam
α (2.5714474995)	-0.79372	-0.79455	-0.79511
2α (1.140290308)	0.28902	0.28922	0.29102
3α (1.431157192)	0.24773	0.25275	0.24975
4α (2.28058062)	-0.56553	-0.57670	-0.57261
5α (0.29086688)	0.56738	0.57935	0.57394
6α (2.86231438)	-0.33861	-0.34405	-0.33719
7α (0.84942342)	0.06305	0.05851	0.05079
8α (1.72202408)	0.10527	0.11646	0.12336
9α (1.98971373)	-0.12430	-0.13432	-0.13804
10α (0.58173377)	0.05502	0.05765	0.05633
11α (3.13000404)	0.01143	0.01623	0.02280
12α (0.55855654)	-0.02549	-0.03245	-0.04237
13α (2.01289096)	-0.00084	0.00349	0.01361
14α (1.69884685)	0.02581	0.02291	0.01545
15α (0.87260065)	-0.02119	-0.01408	-0.01064
16α (2.83913716)	-0.00847	-0.02258	-0.02248
17α (0.26768966)	0.03684	0.05404	0.05222
18α (2.30375784)	-0.04279	-0.05501	-0.05359
19α (1.40797996)	0.02787	0.02854	0.02872
20α (1.16346754)	-0.00885	0.00154	0.00007

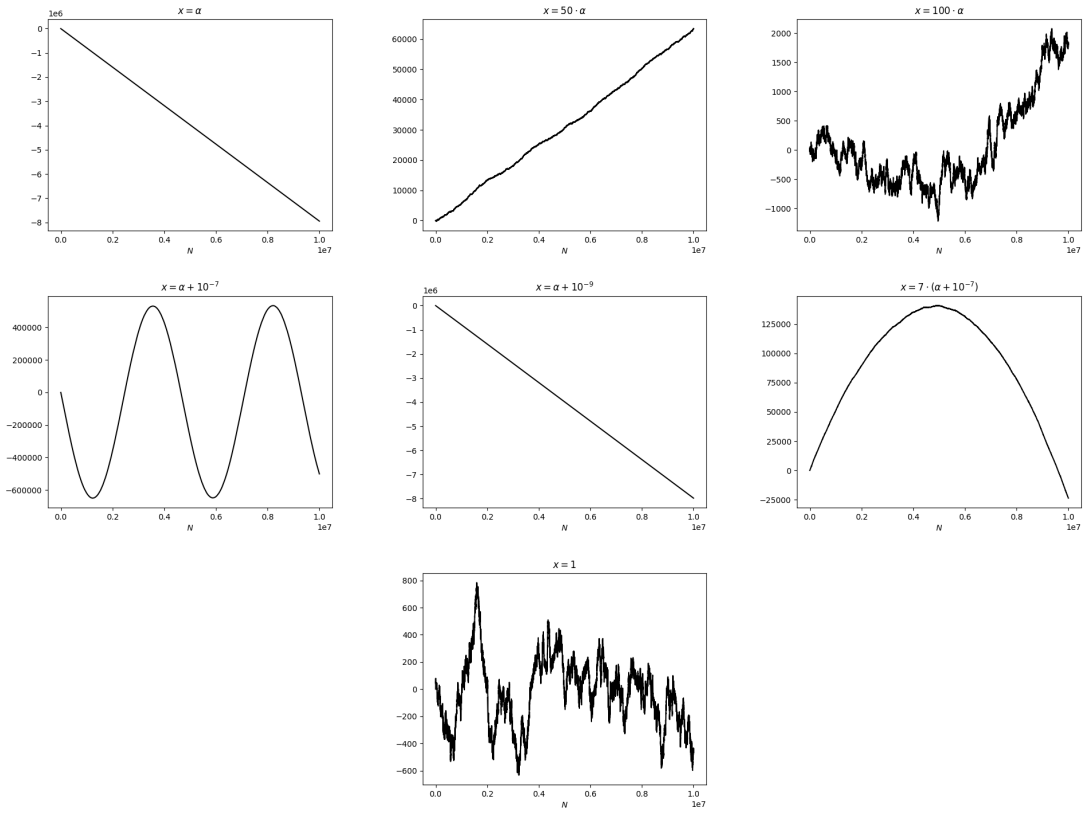

Figure 2: $f_N(x)$ for select fixed values of x and $N \in \{1, \ldots 10^7 - 1\}$.

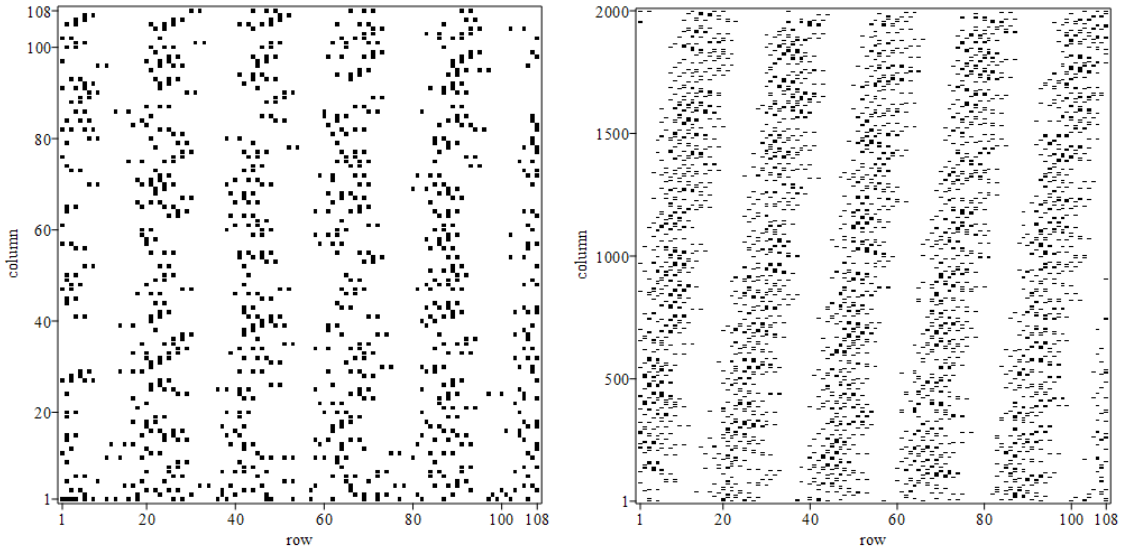

Figure 3: Pixel plots of reduced sets.

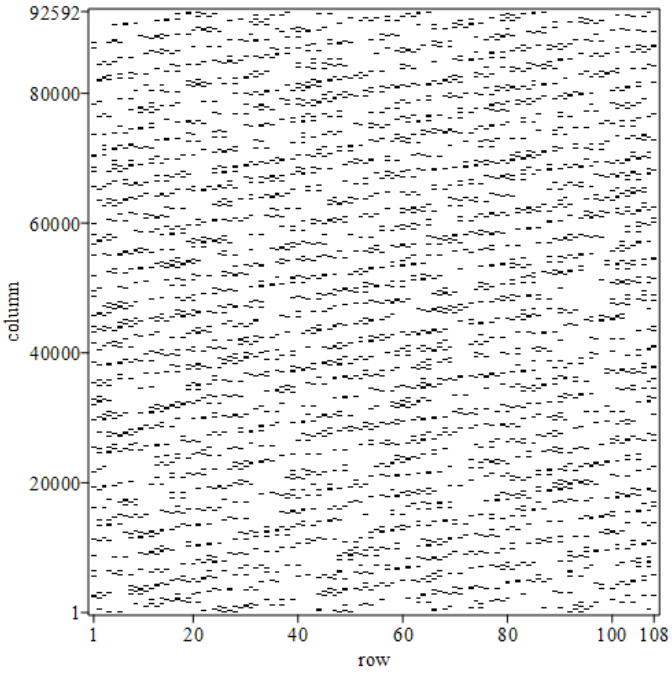

Figure 4: Pixel plot of the first 10^7 Ulam numbers.

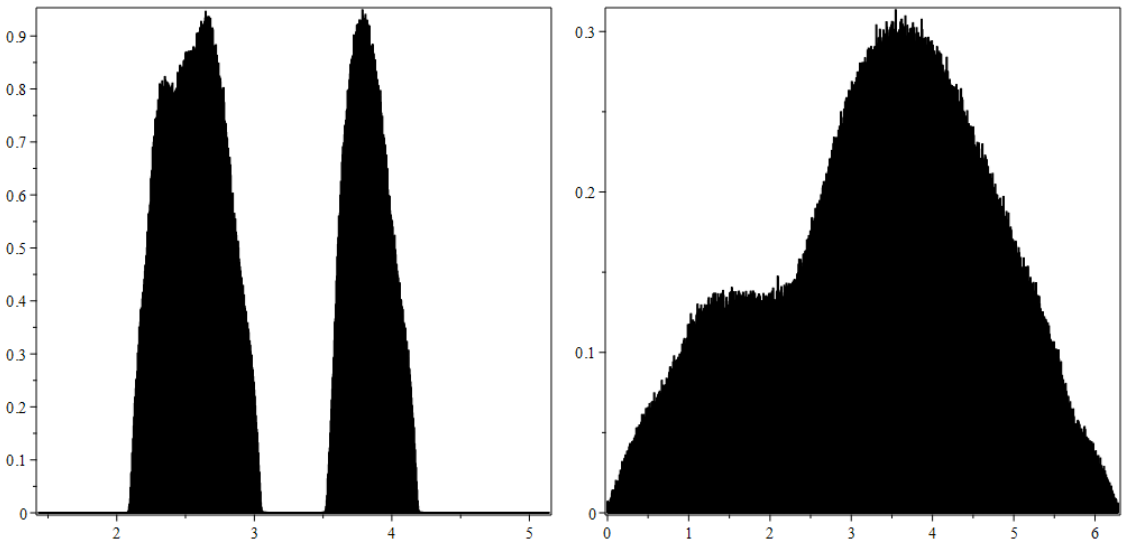

Figure 5: Distribution of S_N for $N = 10^6$ for α and 6α.

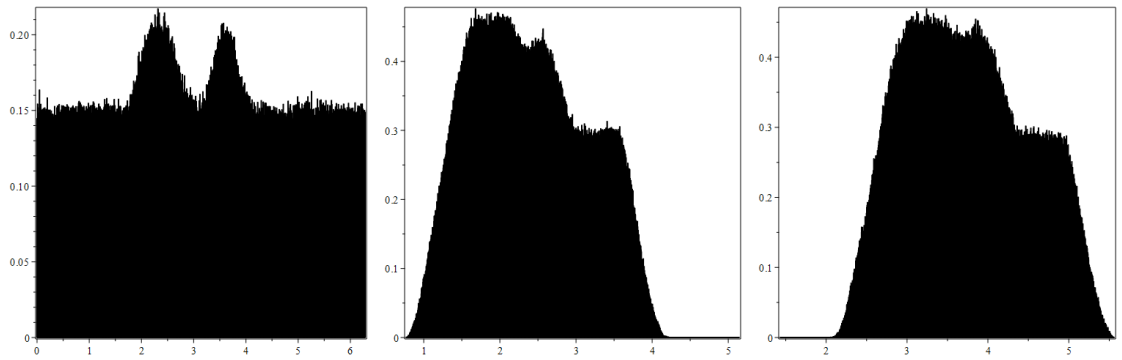

Figure 6: Distribution of S_N for $N = 10^6$ for α rounded up to 7 digits, for $\alpha = 2.5714474, 2.5714476$.

Table 2: Cosine pseudorandom experiment results.

Value	Mean	Approx.
0.2908569968	0.494	5α
2.571436358	-0.489	α
2.862317336	-0.306	6α
1.431150773	0.281	3α
1.431153589	0.275	3α
1.431147311	0.270	3α
2.571410664	0.158	α
1.431165342	0.125	3α
2.862326688	-0.124	6α
0.8494339180	0.112	7α
0.2908077842	0.092	5α
0.8494434892	0.091	7α
0.8494261535	0.082	7α
1.431167819	0.080	3α
0.8493977674	-0.071	7α
2.012877602	-0.065	13α
2.571533226	0.058	α
3.129990560	0.057	11α
0.5817249181	0.056	10α
2.571371618	0.055	α
2.280492787	0.054	4α
2.862396139	0.053	6α
2.571345848	-0.051	α
1.698821692	-0.047	14α
0.2909935426	-0.045	5α
1.698852594	0.045	14α
2.571296220	-0.044	α
2.303754300	-0,043	18α
1.989726789	-0.043	9α
2.303753774	-0.043	18α
1.431087745	0.042	3α
0.2908379101	-0.040	5α
0.2909910223	-0.039	5α
0.2676956441	0.039	17α
0.2914101180	-0.038	5α
2.012862229	-0.037	13α
2.680627864	0.036	?
0.2908164740	0.035	5α

Value	Mean	Approx.
2.280032395	0.034	4α
0.2909260433	0.034	5α
1.698813638	-0.034	14α
1.140363151	-0.033	2α
2.280489584	0.032	4α
2.171397916	0.031	?
2.280664902	0.030	4α
2.862283023	0.030	6α

Table 3: Sine pseudorandom experiment results.

Value	Mean	Approx.
2.571436358	0.531	α
2.280558758	0.405	4α
2.571419379	0.372	α
2.862335205	-0.260	6α
1.431171548	0.251	3α
1.431165342	0.249	3α
0.2908569968	-0.219	5α
1.140293410	0.185	2α
1.431182909	0.131	3α
0.2909350130	0.122	5α
2.571410664	0.119	α
0.2907912295	-0.104	5α
0.8494261535	-0.096	7α
0.5585570850	-0.096	12α
0.2907892342	-0.094	5α
2.571326636	0.093	α
0.2909260433	0.090	5α
2.571323269	0.076	α
1.722052451	0.070	8α
2.862378170	-0.068	6α
1.698845067	-0.063	14α
0.2914010272	0.057	5α
0.2907572099	-0.057	5α
1.140363151	0.052	2α
2.280664902	-0.051	4α
2.571533226	-0.050	α
1.140209638	-0.050	2α
0.2909756705	0.049	5α
1.431087745	-0.048	3α
0.2910320415	0.047	5α
0.2908077842	-0.046	5α
0.2909935426	0.042	5α
2.862433021	-0.041	6α
0.2908084424	-0.041	5α
1.698853038	-0.040	14α
1.431039228	-0.039	3α
0.2913863729	0.039	5α
0.8494339180	-0.038	7α
2.280852761	-0.038	4α

Value	Mean	Approx.
2.570992589	0.037	α
0.2904089143	-0.037	5α
0.5585902345	0.037	12α
2.571601380	-0.036	α
2.862835423	-0.036	6α
2.571750719	-0.036	α
2.571650842	-0.035	α
2.571669804	-0.035	α
0.5590769745	0.033	12α
0.2661285736	0.032	?
1.431271589	0.032	3α
2.280424626	0.032	4α
0.2919850428	0.032	5α
2.893332718	-0.032	?
0.2901417212	0.031	5α
0.5590807799	0.031	12α
2.893329854	-0.031	?
2.571046418	0.031	α
2.012862229	0.030	13α
3.022839268	0.030	?
1.989185592	0.030	9α

References

[1] OEIS A2858

[2] A Hidden Signal In The Ulam Sequence (S. Steinberger, 2016)

[3] The Ulam Numbers up to One Trillion (P. E. Gibbs, 2017)

[4] An efficient method for computing Ulam numbers (P. E. Gibbs, 2015)