Ulam sequences

Helena Jorquera, Ignacio González Mantecón,
Libei Wang and Emil Rugaard Wieser

December 2023

Contents

Acknowledgements

We would like to thank Stefan Steinerberger and Morten Risager for answering questions related to this report.

1 Initial results

1.1 Ulam sequence

The ulam sequence $U_{u,v,n}$ is the sequence of numbers generated by the two initial sequence elements u and v, both positive real numbers, where the following sequence elements are defined recursively as the smallest unique sum of prior elements. Note that as long as u and v are distinct the sequence is infinite as the sum $U_{u,v,n} + U_{u,v,n+1}$ is always unique by the monotonicity of summing up positive numbers. Throughout, $(a_n)_n$ will denote the Ulam sequence corresponding to u, v where u, v will be clear by context.

In "A Hidden Signal in the Ulam sequence"[1] Stefan Steinerberger studies the sequence where $(u, v) = (1, 2)$ by considering the Fourier series with coefficients $U_{1,2,n}$. We closely follow his example and study the function $f = \sum_{n=1}^{\infty} \cos(a_n x)$ for $U_{u,v,n}$. He studies the function by approximating by truncation. He finds that there is a consistent $\alpha_{1,2}$ that gives rise to a local minimum of this function, with a very large negative value. This minimum means that the cosines take negative values very often, hence $\alpha a_n \mod 2\pi$ is very often in the interval $[\frac{2\pi}{4}, \frac{3\cdot 2\pi}{4}]$, giving rise to a sequence that is very much not equidistributed. We seek to generalize this result to other pairs of (u, v).

1.2 Background observations

Commuting u and v results in essentially the same sequence (only the first two entries are commuted), so we may assume $u < v$. If u, v share a common factor, say k with $u = ku'$, $v = kv'$ then $U_{u,v} = k \cdot U_{u',v'}$, by an inductive argument.

1.3 Generating Ulam numbers in general

By simple recursion one may find an algorithm that generates Ulam numbers for arbitrary u, v. One can more or less effectively implement this, but the gist of it is that one simply sums up all previously seen Ulam numbers and checks whether the sums are unique and simply picks the smallest such. Summing up all pairs of integers makes this run in $O(N^2)$ time[2], which is good for preliminary results. With this method, one can in a reasonable time find 100s to 1000s of Ulam numbers.

2 Finding α's

To estimate $\alpha_{u,v}$ we optimize the truncated Fourier series $f_N(x) = \sum_{n=1}^{N} cos(a_n x)$ and find an estimate of the global minimum of this truncated function, $\alpha_{u,v,N}$ Optimizing a sum of trigonometric functions in general is difficult, doing something along the lines of Newton's method will lead one to a local optimum, but it is very sensitive to the initial value as a trigonometric series has many local optima. We get around this by doing a grid search in the interval $[0, \pi]$, computing the function at many x values and finding the lowest function value. This will provide a good guess at an initial value for an optimization. To limit the sources of error we initially do this for low values of N.

2.1 Early results

These methods prove fruitful and we compute candidate values for $\alpha_{u,v,N}$ for $u = 1$ and v an integer with $f_N(\alpha) \approx -0.8N$(as shown in Figure 1). For $u = 2$ and v odd we find $\alpha_{(u,v,N)}$s with $f_N(\alpha_{(u,v,N)}) \approx -0.992N$. For other integer values of u we find for up to 400 pairs u, v an $\alpha_{u,v,500}$ for which $f_{500}(\alpha_{u,v,500}) < -0.64 \cdot 500$ (Examples of $u = 3$ is shown in Figure 2). This seems to suggest that the phenomenon described in the original article [1] is not unique to $(u, v) = (1, 2)$, and that we can use some of the properties found in prior work.

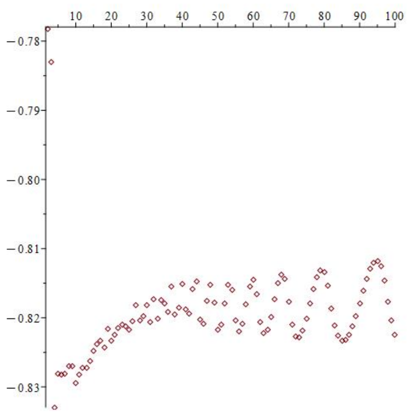

Figure 1: $\frac{f_{500}(\alpha_{1,v,500})}{500}$ for v from 2 to 100, it shows $f_N(\alpha) \approx -0.8N$

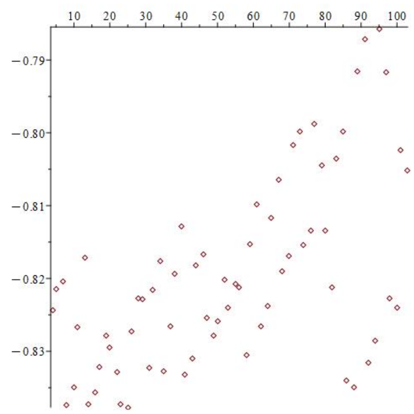

Figure 2: $\frac{f_{500}(\alpha_{3,v,500})}{500}$ for v from 4 to 100, as an example of $f_{500}(\alpha_{u,v,500}) < -0.64 \cdot 500$

2.2 Approximating α

2.2.1 When $u = 1$

Based on our empirical search for $\alpha_{1,v}$ for $v \in \mathbb{Z} \cap [5, 100]$, with $N = 100$ we visually recognize that it looks to have a reciprocal relationship with v, and indeed a power fit returns a fit of $\frac{1.991501017935 20}{v^{0.984248612341787}}$, we interpret this as close to $\frac{2}{v}$, and indeed this approximation seems to, in general, get better as v increases at least as far as the difference from the estimated α is concerned. Notably, this is always an underestimate as well. Figure 3 shows the approximation. The picture is a little more muddled with the quotient, but empirically $1 \leq \frac{\alpha v}{2} \leq 1.10$

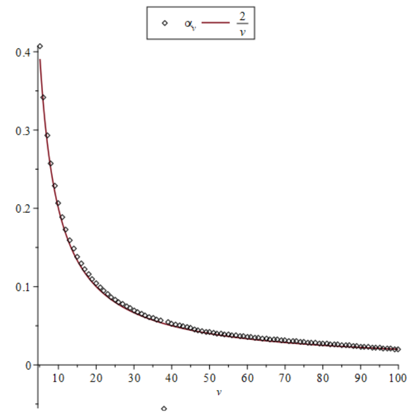

Figure 3: $\alpha_{1,v}$ for v from 5 to 100, compared with $\frac{2}{v}$

Given our presumed rate of convergence we are indeed relatively confident that $\frac{2}{v} \leq \alpha_{1,v} \leq \frac{2.2}{v}$ for all $v \geq 5$. It would be interesting to find out exactly what describes the divergence from these bounds. This was done for $N = 100$ initially in the project with the inefficient algorithm described above and took several hours. With quadratic time, meaning getting significantly more precision within the project deadline is impossible. This motivates looking for ways to find Ulam numbers and αs efficiently

2.2.2 When $u = 3$

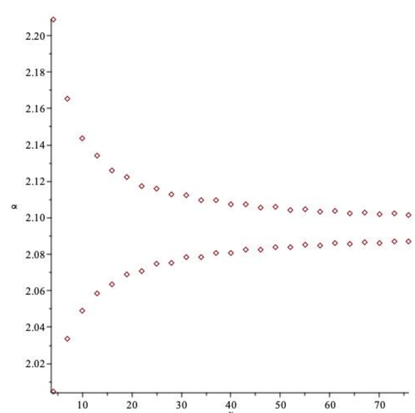

Figure 4: $\alpha_{3,v}$ for v. In the figure, the points corresponding to v being a multiple of 3 are omitted.

Figure4 shows $\alpha_{3,v}$ against v when $v \equiv 1, 2 \pmod 3$. In figure4, the points corresponding to v being a multiple of 3 are omitted. Because according to section1.2, $U_{3,3k} = 3 \cdot U_{1,k}$

We can see that for large enough v we have that $\alpha \approx 2.1$ is a good approximation. e.g. ulam$(3, 100, 1000)$ has $\alpha_{3,100} = 2.0996478790873465$.

2.2.3 When u or v is even

We notice that for u, v coprime and u even, $\alpha \approx \pi$. However, this observation is based on a limited dataset: it was tested for 20 different values of v for each even u from 2 to 20, and for 5 different values of v for each even u from 22 to 40. If this is true, it tells us that for u even and v odd most elements in the sequence $(a_n)_n$ will be odd since $\cos(a_n\pi) = 1$ for a_n even and $\cos(a_n\pi) = -1$ for a_n odd.

For instance, we can compute that the minimum of f when $(u, v) = (4, 7)$ is -492 when using 500 Ulam numbers, which tells us that there are exactly 4 even numbers out of 500 in the Ulam sequence (this result follows from solving $x + y = 500, -x + y = -492$).

Similarly, for u, v coprime, v even and, this time, $u \geq 7$, we have $\alpha \approx \pi$. Again, if this is true, it tells us that most elements in the sequence $(a_n)_n$ will be odd.

3 Finding Ulam numbers given α

The fact that the existence of a local minimum α implicates that Ulam numbers are not equidistributed is used by Gibbs to find Ulam numbers very quickly, conjecturally in linear time[2] (if the phenomenon persists) and experimentally so as well. They produce an algorithm that takes very efficient guesses at Ulam numbers by using approximations of α and the average gap between Ulam numbers, giving correct results regardless of the approximation but increasing in speed with the accuracy. The algorithm works by sorting Ulam numbers by residues modulo $\lambda = \frac{2\pi}{\alpha}$, using the unequal distribution of the Ulam numbers to quickly determine whether a given number is an element of the sequence. Our preliminary work suggests that these speed-ups can apply to our case as well, we've modified their code to work to do so. The algorithm is indeed very fast experimentally and allows us to compute a lot more Ulam numbers

3.1 Optimizing at large N's

A naive optimization of a very large sum of cosines with a grid search runs into the trouble described with there being a large number of minima, and therefore either a very fine grid is required (meaning the function must be evaluated a large number of times) or the risk of getting a bad initial value is high. In our case, however, the trigonometric series are interrelated and the very conjecture we are testing is that the x-intercept of the minima converges. Hence optimizing at small N gives a guess at the x-intercept at large N. In private communication Stefan Steinerberger claims his computations suggest that the rate of convergence is around $O(\frac{1}{N})$. On the other hand, the number of local minima of f_N on a given interval I with length L looks visually to be around $O(NL)$. Hence if Steinerberger is right we can without trouble restrict the interval of our grid search to an interval of length $O(\frac{1}{N})$ around α_N and in and in this interval, we have $O(\frac{N'}{N})$ minima of $f_{N'}$. If we at each step pick $N' = cN$ we obtain a constant amount of minima in our interval and can therefore pick a fixed grid size and run the same risk of picking a bad initial value. This means that the runtime of our optimization only scales with the increased number of terms and not with the amount of function calls. Experimentally this rate of convergence has not been contradicted for other values of u and v. Now this runs the risk of being circular, as we are trying to find minima conjectured to exist and we assume to be near them, but we simply qualify this by checking that $f_N(\alpha_{u,v,N}) \leq -0.64N$

3.2 An algorithm that produces Ulam numbers and alphas efficiently concurrently

First, calculate the first 10^2 Ulam numbers using any method and compute the corresponding $\alpha_{u,v,N}$ and the average gap size. Then feed this back into the modified Knuth-Gibbs algorithm and find 10^3 sequence elements and a corresponding $\alpha_{u,v,10^3}$, do this iteratively, finding 10^{n+1} numbers with the precision given at 10^n until the desired number of elements has been found and the desired accuracy of α to n'th decimal has been found. If knowing $\alpha_{u,v}$ to an approximated value only gives a constant time speed decrease (this is very conjectural) this gives a running time on the order of $O(n\log(n))$ as we must run the algorithm for each power of 10 less than the desired one. This algorithm will allow us to test our hypotheses for the values of α for a large number of initial values, but due to technical issues relating to passing between the algorithms, we have not done so at the time of the preliminary report. Finding Ulam numbers is done in Java as we run modified code by Gibbs, and optimization is done in Julia.

4 Equidistribution

Now we investigate whether the sequence $(x_n)_n := (a_n\alpha/2\pi)_n$ is equidistributed. Since $f_N(\alpha_{u,v}) = \sum_{n=1}^{N} \cos(a_n\alpha) \leq -0.64N$ shown in section 2.1, we know that at least 64% of the cosines are negative. In turn, this implies that at least 64% of $a_n\alpha$ are inside the interval $[\pi/2, 3\pi/2]$ mod 2π. Then 64% of $\langle a_n\alpha/2\pi\rangle$ are inside the interval $I := [1/4, 3/4]$. This is already more than 50% so we expect the sequence to not be equidistributed.

We know that for $(u,v)=(1,2)$ there are 4 outcasts, that is, 4 elements $\langle x_n \rangle$ outside of I. We write a function that lists and counts for each pair of elements (u,v) the elements in the sequence $(\langle x_n \rangle)_n$ which are outside of the interval $I=[1/4,3/4]$, that is, it computes the list $L_{u,v} := [x_i \in (x_n)_n : \langle x_i \rangle \notin I]$. Out of 400 pairs u,v (letting $u=1,...20$ and $v=u+1,...,u+20$) the function tells us that 146 pairs have less than 5 elements outside of I. So, although none of the sequences are equidistributed, some sequences have more than 4 outcasts, e.g. the sequence for $(u,v)=(9,17)$ has 48 outcasts out of 500.

4.1 When u or v is even

Now let u be even and v coprime to u. In this case, we suspect that $\alpha \approx \pi$, in which case the sequence $(x_n)_n$ is $(a_n/2)_n$. If this is true, $\langle x_n \rangle = 0.5$ for most n and $\langle x_n \rangle = 0$ for the rest since, as observed in Section2.2.3, most of the elements in the sequence $(a_n)_n$ are odd. In particular, $\langle x_1 \rangle = 0 \notin I$.

Similarly, let v be even and coprime to $u>5$. In this case, we suspect $\alpha \approx \pi$ so that the sequence $(x_n)_n$ is $(a_n/2)_n$. If this is true, $\langle x_n \rangle = 0.5$ for most n and $\langle x_n \rangle = 0$ for the rest since, as we observed earlier in Section2.2.3, most of the elements in the sequence $(a_n)_n$ are odd. In particular, $\langle x_1 \rangle = 0.5 \in I$.

4.2 Cases we ignore

We notice that $L_{u,v} = L_{ku,kv}$ where k is an integer, so we restrict our attention to u,v coprime.

From looking at the output we see that very often $\langle x_1 \rangle = 0 \notin I$. We write a function which returns the pairs (u,v) for which $\langle x_1 \rangle \in I$. As expected, it returns all pairs (u,v) with v even and coprime to u, and these are in fact the vast majority, but it also returns a few other pairs. Since we already understand $(x_n)_n$ very well when v is even we focus on those pairs for which $\langle x_1 \rangle = 0 \notin I$.

4.3 A pattern in the lists of outcasts of $(x_n)_n$

We now restrict our attention to the cases where the two properties above are satisfied, that is, where u,v are coprime and where $\langle x_1 \rangle \notin I$. We notice that the sequence of second values in each list $L_{u,v}$ seems to follow a pattern, the position of the outcasts seem to increase constantly as we increase v for a fixed u.

Let $p_{u,k}$ be the k^{th} number coprime to u and greater than u for which $\langle x_1 \rangle \notin I$. Then, for instance, for $u=2$, the second element l_2 in the list $L'_{2,v}$ is given as follows:

when $v = p_{2,k}$**,** $l_2 = \langle x_{k+5} \rangle = 0$, that is, for $u=2$,

when $v = 5,\ l_2 = \langle x_6 \rangle = 0$,
when $v = 9,\ l_2 = \langle x_8 \rangle = 0$,
when $v = 11,\ l_2 = \langle x_9 \rangle = 0$, etc.

For $u = 3$ we have something very similar. When $v = p_k$, $l_3 = x_{k+4} = 0$ if k is odd and $l_3 = x_{k+3} = 0$ if k is even, that is,
when $v = 4,\ l_3 = \langle x_5 \rangle$,
when $v = 5,\ l_2 = \langle x_5 \rangle$,
when $v = 7,\ l_2 = \langle x_6 \rangle$,
when $v = 8,\ l_2 = \langle x_6 \rangle$,
when $v = 10,\ l_2 = \langle x_7 \rangle$,
when $v = 11,\ l_2 = \langle x_7 \rangle$.

In this case, this means that, as v increases by 2, the position of the outcast moves to the right by one position:

when $v = 4$, $[\langle x_1 \rangle, ..., \langle x_8 \rangle] = [0.055, 0.406, 0.461, 0.516, \underline{\mathbf{0.867}}, 0.571, 0.274, 0.625]$
when $v = 5$, $[\langle x_1 \rangle, ..., \langle x_8 \rangle] = [0.957, 0.595, 0.553, 0.51, \underline{\mathbf{0.148}}, 0.467, 0.424, 0.744]$
when $v = 7$, $[\langle x_1 \rangle, ..., \langle x_8 \rangle] = [0.034, 0.412, 0.446, 0.48, 0.514, \underline{\mathbf{0.859}}, 0.548, 0.582]$
when $v = 8$, $[\langle x_1 \rangle, ..., \langle x_8 \rangle] = [0.971, 0.589, 0.561, 0.532, 0.503, \underline{\mathbf{0.15}}, 0.474, 0.445]$
when $v = 10$, $[\langle x_1 \rangle, ..., \langle x_8 \rangle] = [0.024, 0.412, 0.435, 0.459, 0.483, 0.506, \underline{\mathbf{0.847}}, 0.53]$
when $v = 11$, $[\langle x_1 \rangle, ..., \langle x_8 \rangle] = [0.978, 0.587, 0.566, 0.544, 0.523, 0.501, \underline{\mathbf{0.153}}, 0.479]$

Looking at the original Ulam sequences for $u = 3$ we see that the position of the outcast in $L_{3,v}$ corresponds to the position where the sequence $U_{3,v}$ differs from the sequence $[3, v, 3+v, 6+v, 9+v, ...]$:

$U_{3,4} = [3, 4, 7, 10, \underline{\mathbf{11}}, 13, 15, 16]$
$U_{3,5} = [3, 5, 8, 11, \underline{\mathbf{13}}, 14, 17, 18]$
$U_{3,7} = [3, 7, 10, 13, 16, \underline{\mathbf{17}}, 19, 22]$
$U_{3,8} = [3, 8, 11, 14, 17, \underline{\mathbf{19}}, 20, 23]$
$U_{3,10} = [3, 10, 13, 16, 19, 22, \underline{\mathbf{23}}, 25]$
$U_{3,11} = [3, 11, 14, 17, 20, 23, \underline{\mathbf{25}}, 26]$

More generally, for a fixed u in $[1, ..., 20]$ we have that as v increases by k for some k, the position of the outcast moves to the right by one position.

u	v	outcast
2	5	x_6
	7	x_7
	9	x_8
	11	x_9
	13	x_{10}
	15	x_{11}
3	4	x_5
	5	x_5
	7	x_6
	8	x_6
	10	x_7
	11	x_7
4	5	x_5
	7	x_5
	9	x_6
	11	x_6
	13	x_7
	15	x_7

5 Further work

Obtaining a closed formula for $\alpha_{u,v}$ when it exists is of course of theoretical interest, but if one may simply compute efficiently independently of finding Ulam numbers one could potentially speed up Gibbs' algortihm significantly, as it would mean running it once instead for every power of 10 lower than the desired number, making the algorithm of order $O(n)$ instead of $O(nlogn)$. Our statistical work seems to suggest that there are significant relationships between different $\alpha_{u,v}$ and a higher precision may uncover these and predict specific values.

Gibbs' algorithm also relies on the average gap between Ulam numbers. We have not done much systematic work on this problem.

Further simply repeating all the experiments described above for larger N with our much faster algorithm is desired.

References

[1] Stefan Steinerberger. A Hidden Signal in the Ulam sequence, July 2016. arXiv:1507.00267 [cs, math].

[2] Philip Gibbs. An Efficient Method for Computing Ulam Numbers. August 2015.