An innovative integer distribution using Rotor-Router

Jean-Pierre Coursault

October 26, 2025

Abstract

In this paper, we study the operation of the Rotor-Router and attempt to find a method, a process, that allows integer grouping. We begin by defining the mapping $f : \mathbb{N}^* \longrightarrow \mathbb{Z}^2$, which associates to each integer $n \in \mathbb{N}^*$ a position, $f(n) \in \mathbb{Z}^2$, in the plane, determined by the dynamics of the Rotor-Router. The mapping f thus provides a visual representation of the positive natural numbers in the plane. Filters are then applied in order to retain only the integers possessing certain characteristics, such as divisibility, congruence, primality, etc. In other words, starting from $n \in \mathbb{N}^*$, we select subsets satisfying properties such as $d \mid n$, $n \equiv a \pmod{m}$, n prime, ... The Rotor-Router mechanism produces an organization of integers whose observed regularity suggests an unexplored arithmetic distribution. In particular, the study of arithmetic progressions $4k+1$ and $4k+3$, and more generally of congruence classes $n \equiv a \pmod{b}$, has made it possible to highlight non-uniform distribution phenomena, revealing the existence of remarkable arithmetic biases in the distribution of prime numbers.

MSC2020 : 11A41, 05C81
keywords : Prime numbers, Arithmetic sequences, Rotor-Router

1 The Square Rotor-Router

1.1 Description

The rotor-router is a deterministic analogue of a random walk on a graph. In this study, we work on a directed graph. At each vertex v, a cyclic order of its neighbors (or of its outgoing edges) is fixed, and a rotor points to one of them.

Rotor walk rule: each time a particle visits v, the rotor is advanced by one step in the cyclic order, and then the particle is sent along the newly indicated edge (the "pre-rotation" convention).

This mechanism balances the use of directions in the long run: each vertex "serves" its neighbors in turn, with bounded discrepancies — a form of "disciplined randomness" [1].

We will use the aggregation variant (rotor-router aggregation): chips are emitted from a source; each chip follows the rotor walk until it encounters an unoccupied vertex, where it stops and "occupies" that vertex. In this way, one obtains a cluster whose limiting shape in 2D is (numerically and partly theoretically) close to a disk.

Brief history: the mechanism was popularized by James Propp [2] under the name rotor-router; it is related to Eulerian walkers studied in the 1990s [3].

1.2 Rotor-Router mechanism

To better understand the path of an chip from the origin, here are the first 10 steps:

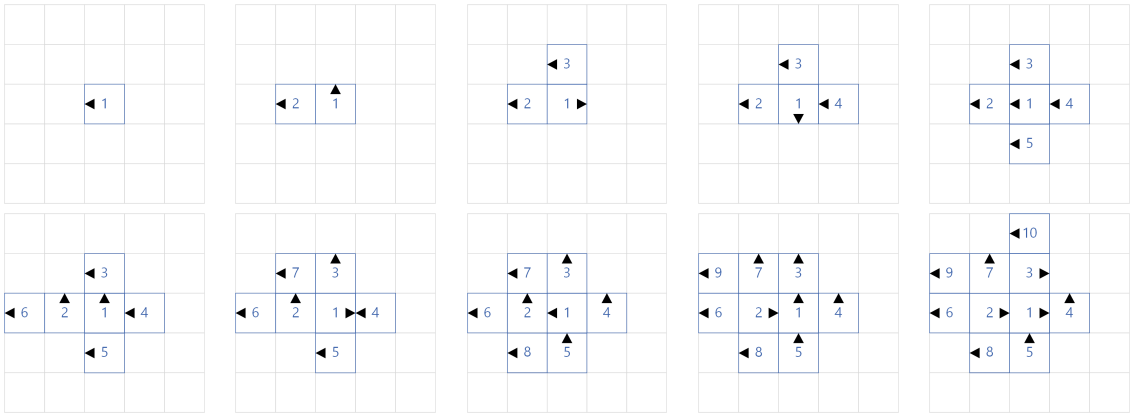


Figure 1: The initial placement steps

The author has chosen, by convention, the clockwise direction for the rotation at each vertex, as well as the order LURD.

- L for Left
- U for Up
- R for Right
- D for Down

Remark. Each vertex of this Rotor-Router has 4 neighboring vertex. We will see in section 3 another type of Rotor-Router.

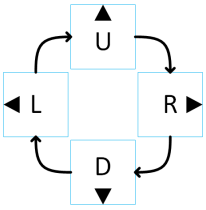


Figure 2: Direction of Rotation

If we assign a color to each direction, for example:

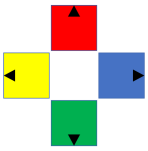


Figure 3: One color per direction

The following figure, was generated for integers $N = 28 \cdot 10^6$.

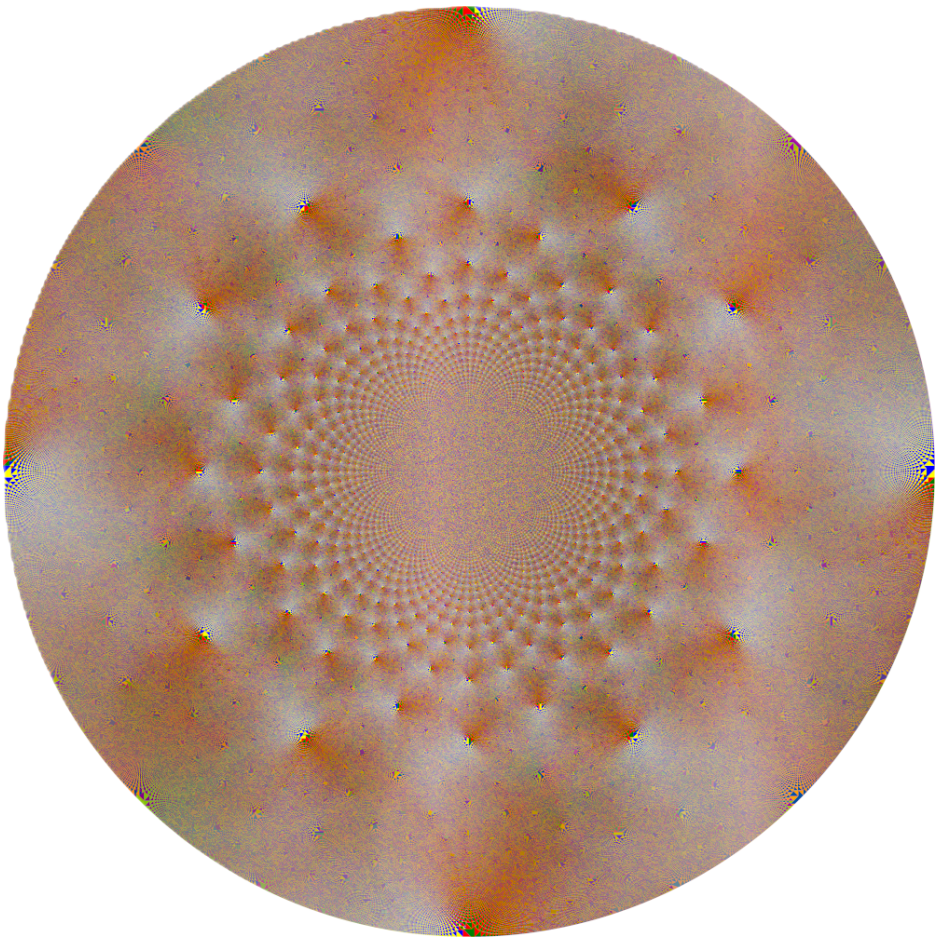
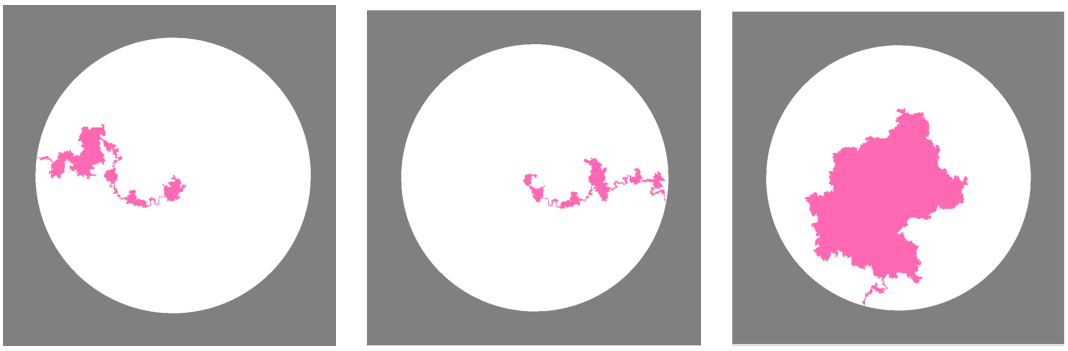


Figure 4: Rotor-Router Pattern Related to the Orientation of Each Cell

1.3 The Path Taken by the chip and Performance

It has been shown [5] that the chip always manages to find an empty site after following a path of varying length. Below is the representation of three paths traced by the agents:



Chip 500008 Chip 500012 Chip 500031

Figure 5: Different paths

Before reaching the outside of the disk, the chip may follow a path whose length can exceed the number of agents already placed. Nevertheless, the average path length m grows linearly with respect to the number n of agents added.

$$m \approx \frac{n}{\pi}$$

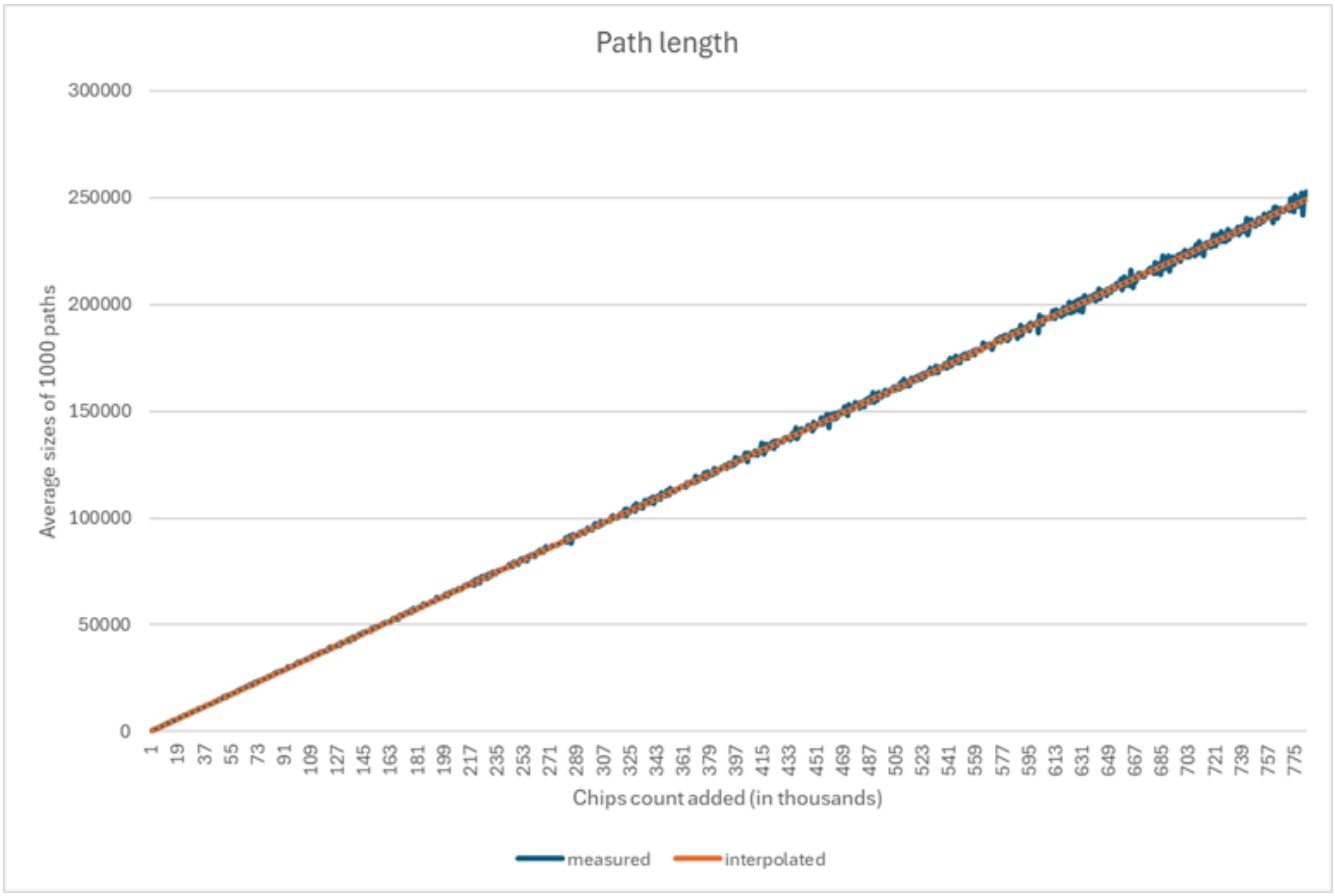


Figure 6: Path Length Proportional to the chip Index

The cost of the n-th chip is (with small oscillations due to lattice effects and boundary roughness)

$$\tau_n = \frac{n}{\pi} + O(\sqrt{n}).$$

The total work required to place N chips is

$$T(N) = \sum_{n \le N} \tau_n \sim \sum_{n \le N} \frac{n}{\pi} = \frac{N^2}{2\pi} \qquad (\Theta(N^2)).$$

The larger the cluster, the more expensive each additional chip becomes. The algorithm for placing the chips is simple and written in C++ (see the pseudo-code in Appendix C). The computer used in this work allows the placement of N chips according to the formula

$$T(N) = 5.767 \times 10^{-10} N^2 \text{ seconds.}$$

In the remainder of this work, the author limited himself to placing 28 million chips, which corresponds to a little more than 4 days of computation.

1.4 Integers in the Rotor-Router Cluster

As shown in Figure 1, the placement of each chip, indexed by its order of placement, defines the integer associated with each site. It is therefore possible to apply filters in order to visualize the distribution of integers according to these filters.

To analyze the angular distribution of the integer points in the Rotor-Router cluster, the disk centered at the origin is partitioned into k sectors of equal opening angle $2\pi/k$, and the number of points in each sector is counted. Plotting these values as a curve as a function of the sector, varying over 2π starting from $-\pi/2$ in the clockwise direction, we obtain a spectrum that expresses the density of the filtered integers. The vertical axis corresponds to the proportion (in %) of integers observed in the sector.

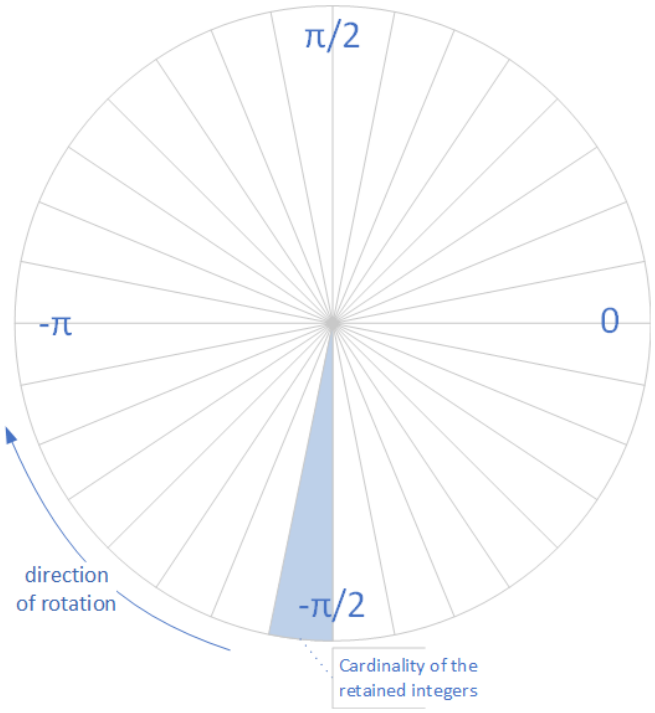


Figure 7: Angular distribution

Remark. As shown in Figure 2, the position of an integer in a quadrant depends on the direction of rotation and the initial orientation of each chip.

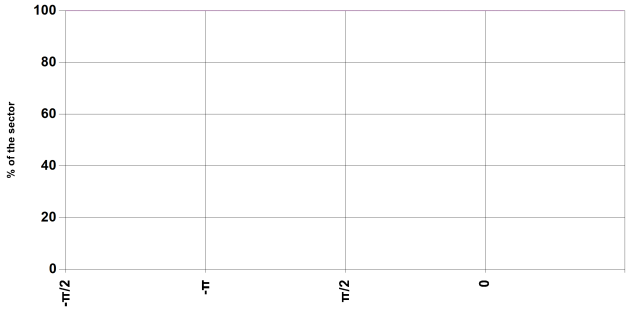


Figure 8: Angular Spectrum

The images presented in the remainder of the document are in PNG format. A white or colored pixel corresponds to a integer retained by the filter, otherwise it is black.

2 Observed Results on the Square Rotor-Router

2.1 Divisibility Filter

2.1.1 Filter $n = 2k$

Let us begin by filtering the integers of the form $n = 2k$ with $k \in \mathbb{N}^*$.

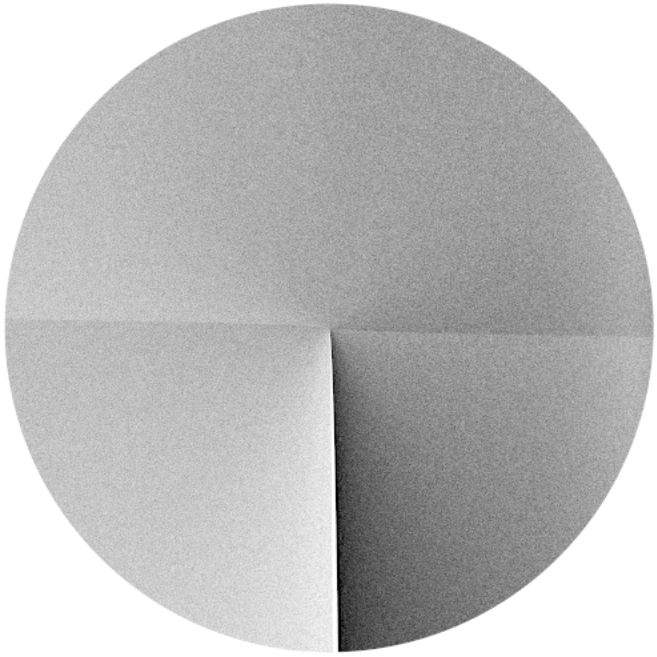


Figure 9: Integers of the form $n = 2k$

A zoom on the lower part of the figure around $-\pi/2$ highlights four densities d_i.

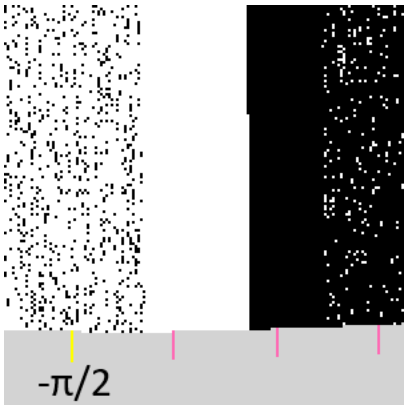


Figure 10: Zoom: Integers of the form $n = 2k$

From left to right:
$d_1 > 85\%, \qquad d_2 = 1, \qquad d_3 = 0, \qquad d_4 < 15\%$

Remark. The change in density between d_2 and d_3 is not aligned with $-\pi/2$; there is a slight shift of approximately $\pi/100$ in the present case. In the remainder of the document we will denote this by $\delta\theta$.

This shift can be observed in the corresponding spectrum:

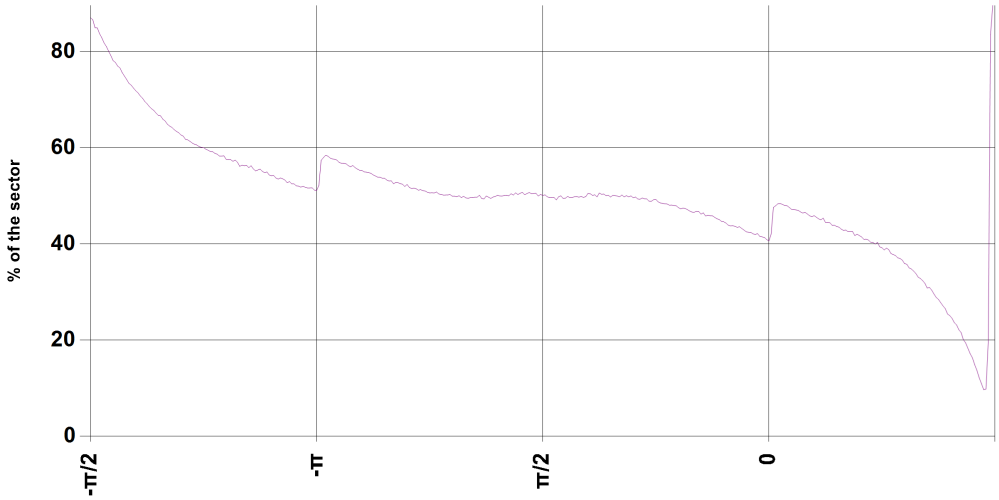


Figure 11: Integers of the form $n = 2k$

Numerical Observation 1. *The density of integers of the form $n = 2k$ with $k \in \mathbb{N}^*$, determined by the dynamics of the Square Rotor-Router, globally decreases starting from $-\pi/2 + \delta\theta$, over an interval of 2π, following the clockwise direction.*

Numerical Observation 2. *Apart from the abrupt change in density at $-\pi/2$ ($+\delta\theta$), three other jumps occur respectively at $-\pi$, $\pi/2$, and 0.*

2.1.2 Filter $n = 2k - 1$

Let us filter the integers of the form $n = 2k - 1$ with $k \in \mathbb{N}^*$. We should obtain a result complementary to the previous one.

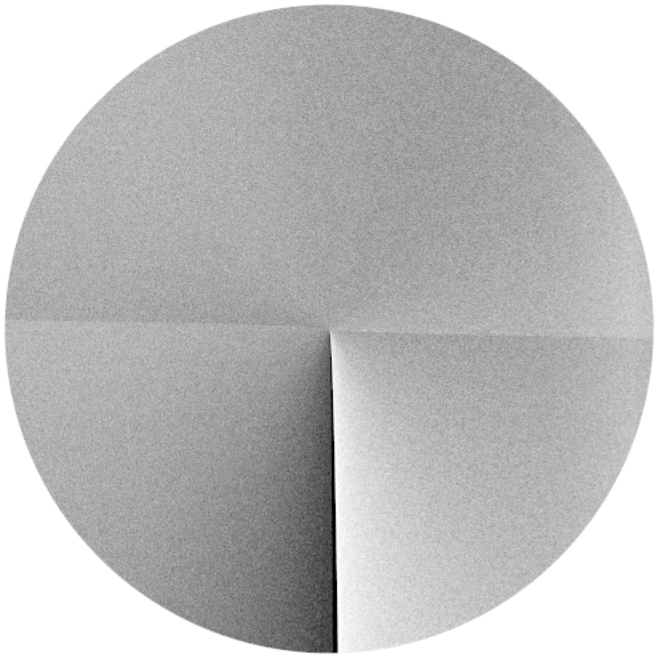


Figure 12: Integers of the form $n = 2k - 1$

Zoom on the lower part of the figure around $-\pi/2$.

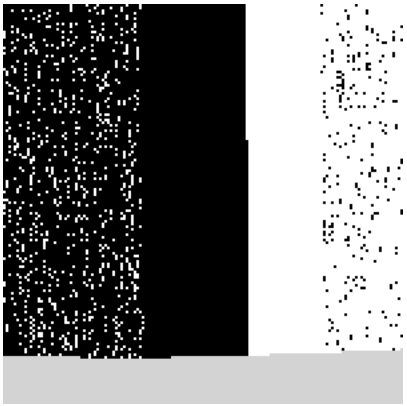


Figure 13: Zoom: Integers of the form $n = 2k - 1$

The corresponding spectrum is as follows:

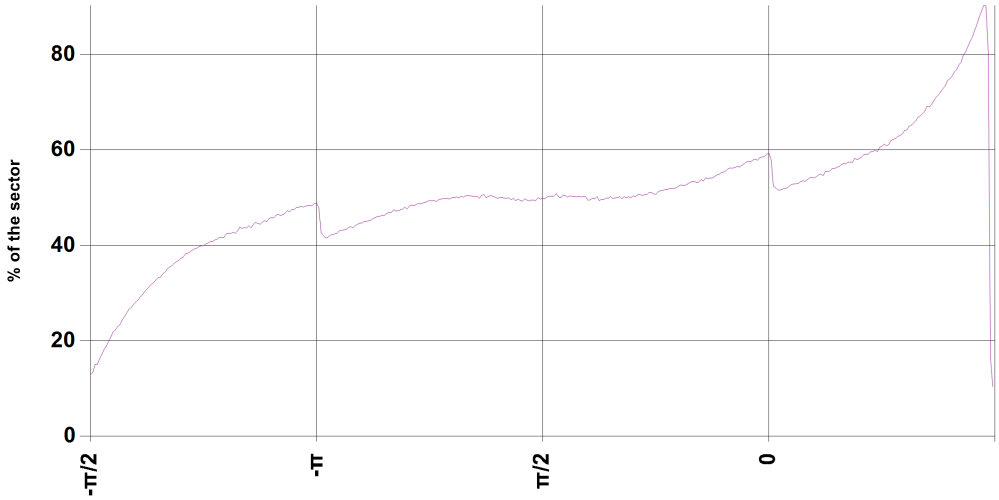


Figure 14: Integers of the form $n = 2k - 1$

Numerical Observation 3. *The density of integers of the form $n = 2k - 1$ with $k \in \mathbb{N}^*$, determined by the dynamics of the Square Rotor-Router, globally increases starting from $-\pi/2 + \delta\theta$, over an interval of 2π, following the clockwise direction.*

2.1.3 Filter $n = 3k$

Let us now consider the integers of the form $n = 3k$ and, more generally, the integers of the form $n = (2m + 1)k$ with $m \in \mathbb{N}^*$.

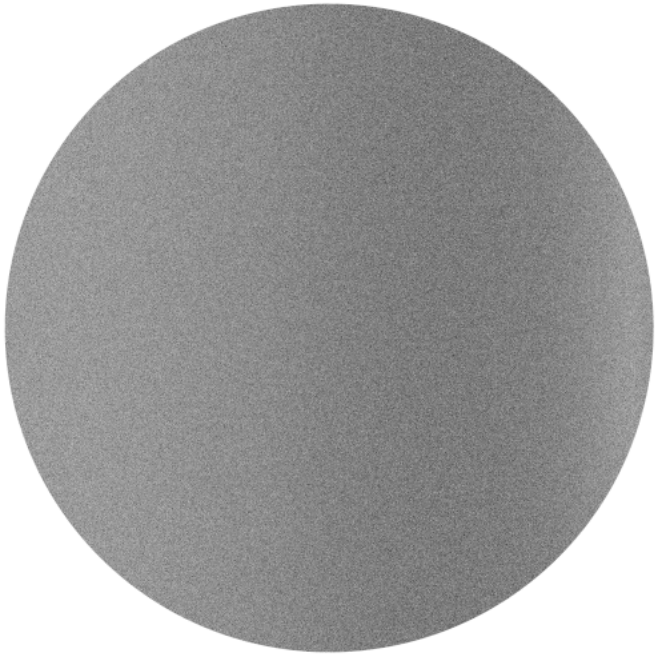


Figure 15: Integers of the form $n = 3k$

Numerical Observation 4. *The density of integers of the form $n = (2m + 1)k$ with $m \in \mathbb{N}^*$, determined by the dynamics of the Square Rotor-Router, is globally homogeneous starting from $-\pi/2$, over an interval of 2π, following the clockwise direction.*

The corresponding spectrum is as follows:

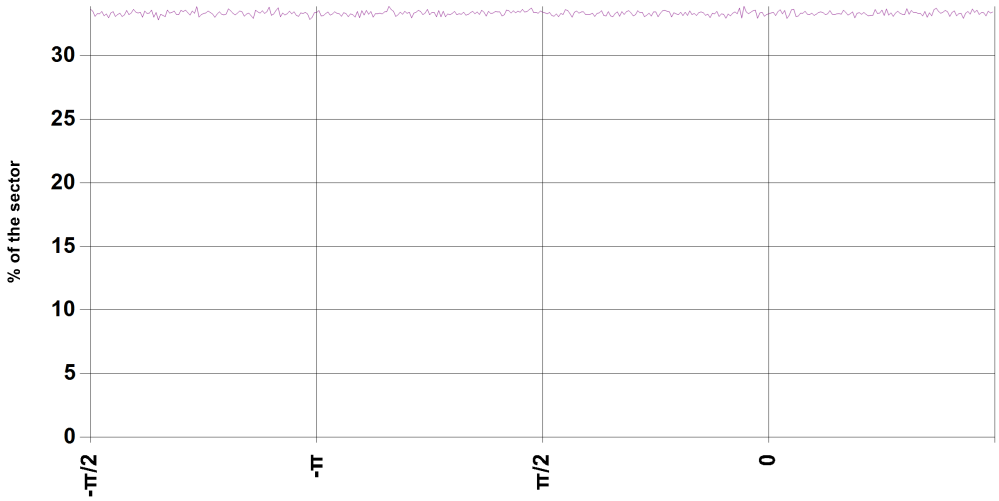


Figure 16: Integers of the form $n = 3k$

2.1.4 Filter $n = 4k$

Those of the form $n = 4k$:

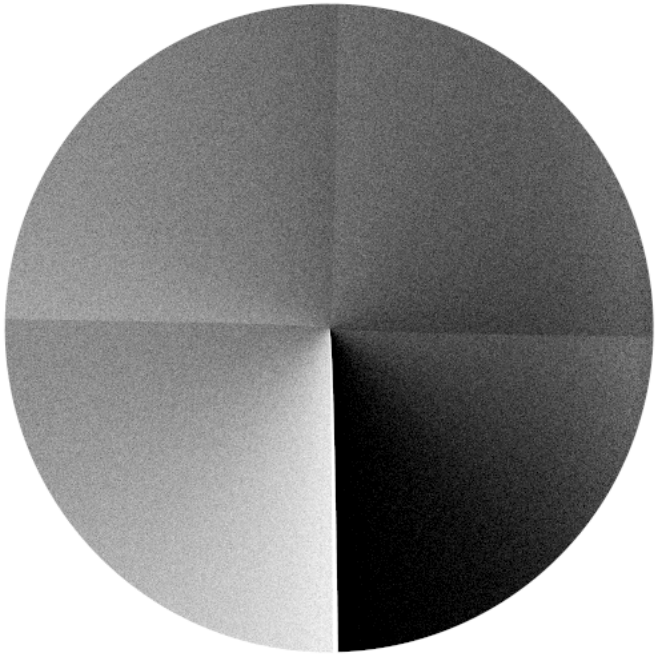


Figure 17: Integers of the form $n = 4k$

The corresponding spectrum is as follows:

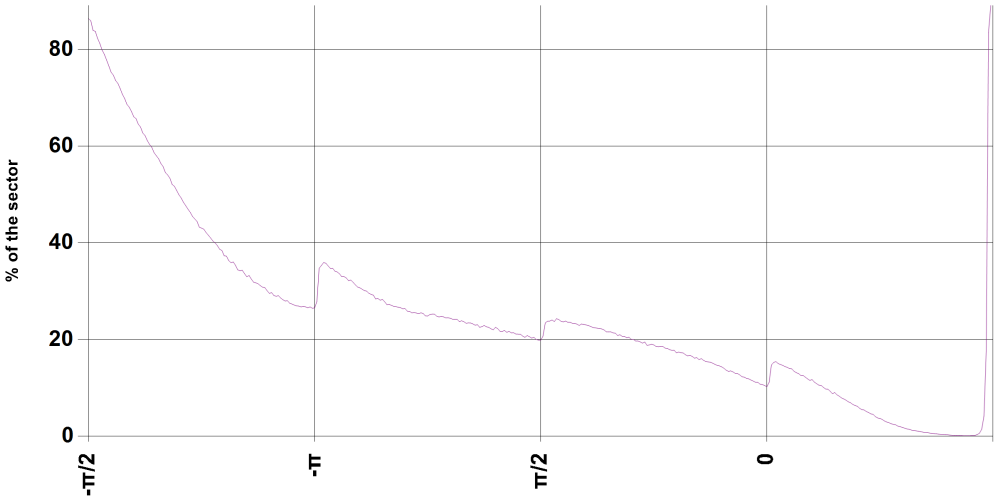


Figure 18: Integers of the form $n = 4k$

We find a distribution similar to that of Figure 9, but the density range is more pronounced for $n = 4k$.

Remark. For the curves above, the disk was subdivided into 400 sectors. The small oscillations observed result from misassignments of an integer to a sector, with an amplitude inversely proportional to the sector size. To illustrate this phenomenon, we also used subdivisions into 100 and 2000 sectors for integers divisible by 4. The overall shape of the curve nevertheless gains in accuracy as the number of sectors increases, in particular for the discontinuities observed at π, $\pi/2$, and 0.

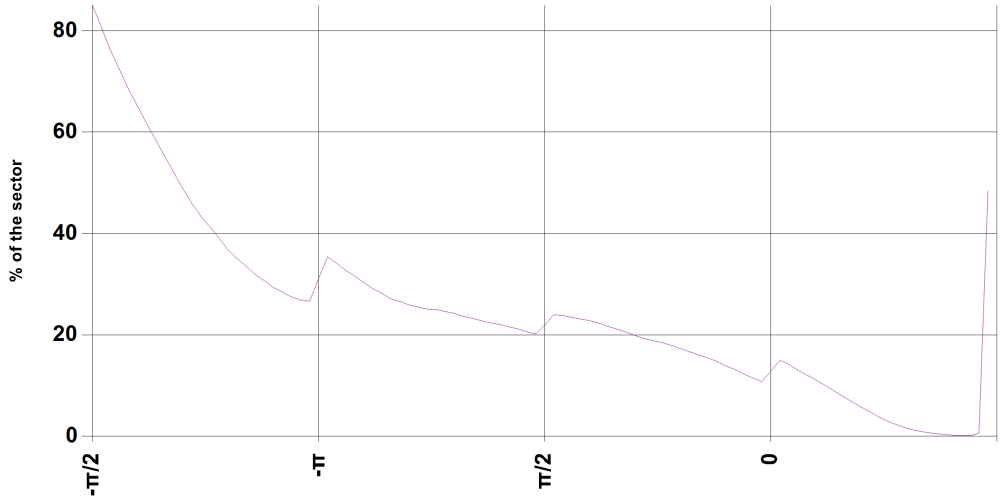


Figure 19: Integers divisible by 4 — 100 sectors

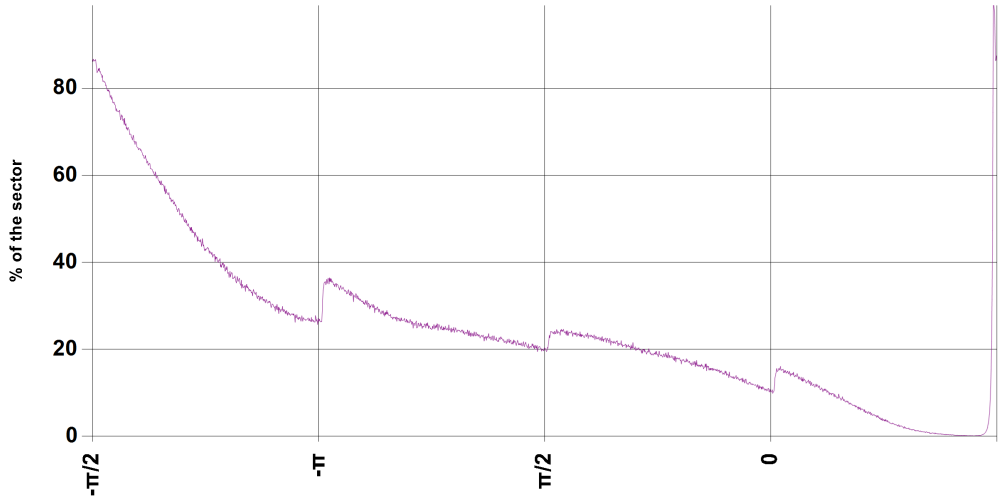


Figure 20: Integers divisible by 4 — 2000 sectors

2.2 Congruence

Let us consider the integers n of the form $n \equiv 3 \pmod 4$ and then $n \equiv 1 \pmod 4$.

In Appendix A are shown the spectra of some congruence classes $b \pmod a$ with $\gcd(b, a) = 1$.

2.2.1 $n \equiv 3 \pmod 4$

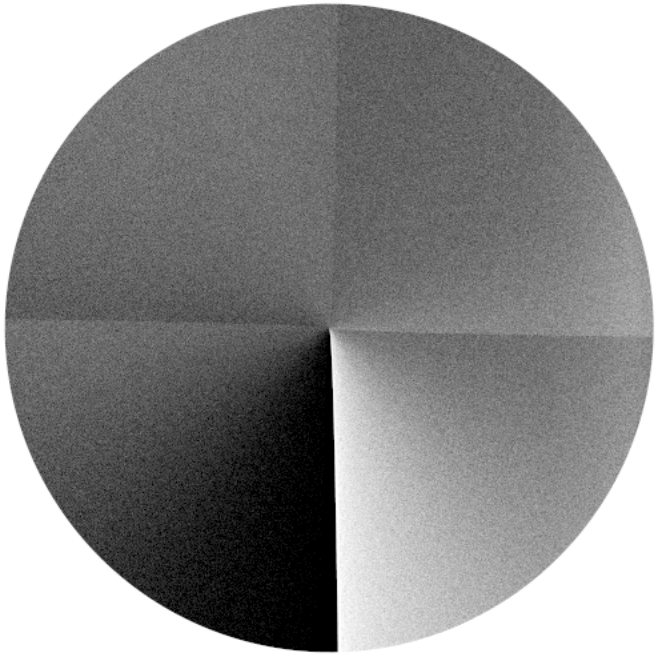


Figure 21: Integers of the form $n \equiv 3 \pmod 4$

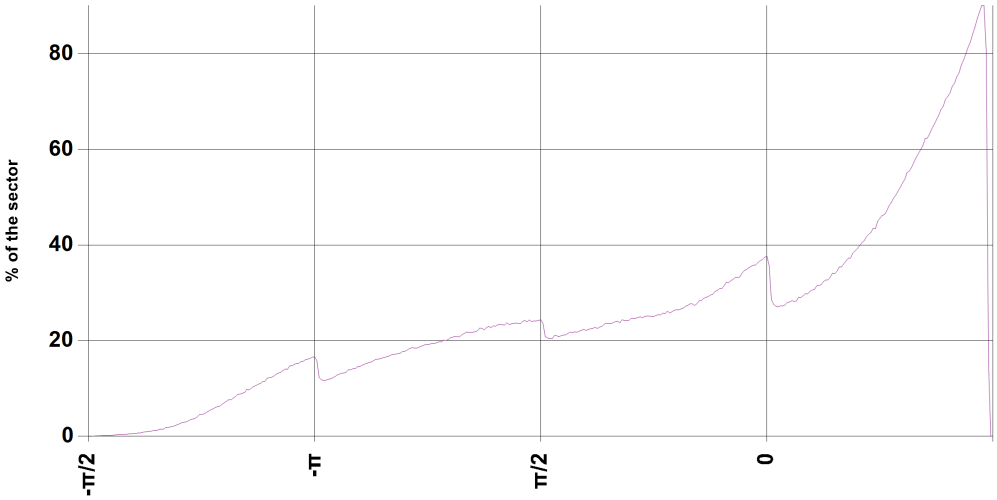


Figure 22: Spectrum of integers $n \equiv 3 \pmod 4$

2.2.2 $n \equiv 1 \pmod 4$

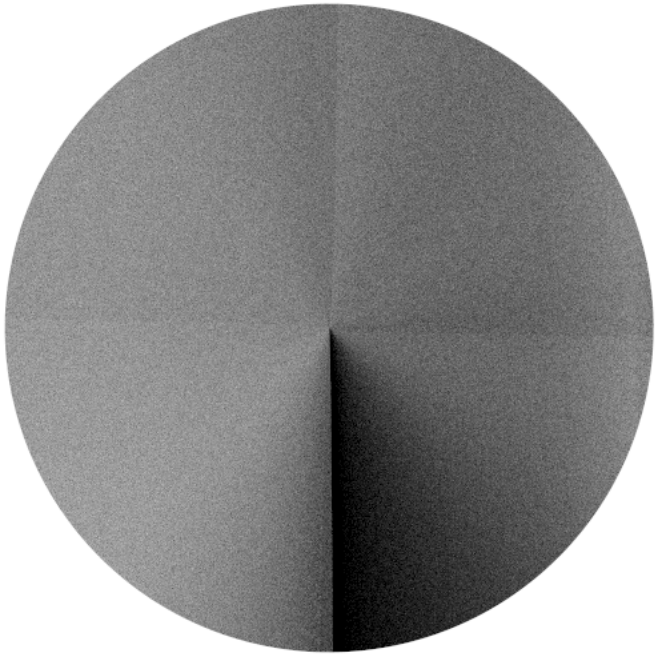


Figure 23: Integers of the form $n \equiv 1 \pmod 4$

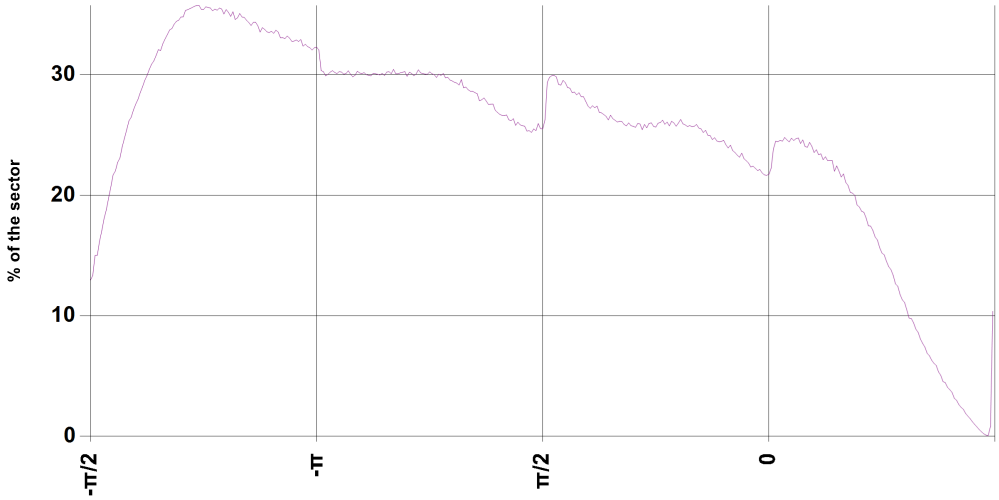


Figure 24: Spectrum of integers $n \equiv 1 \pmod 4$

2.3 Prime Integers

If we retain only the prime numbers:

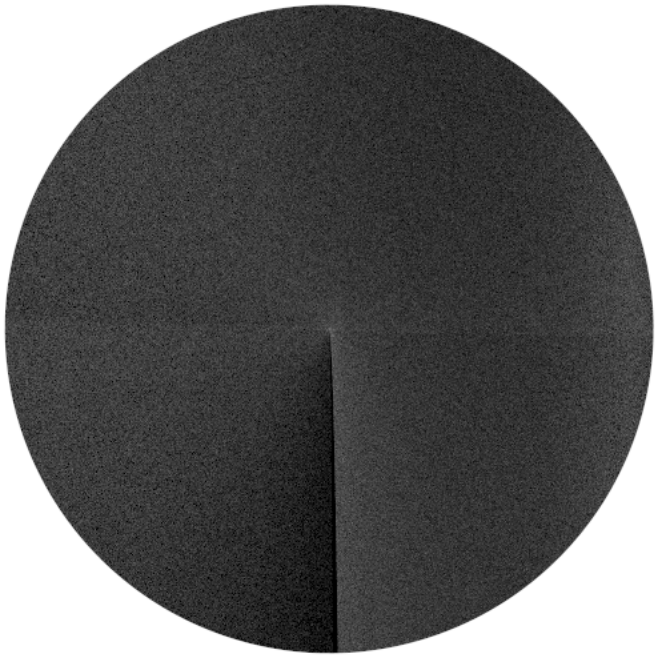


Figure 25: Integers of the form n prime

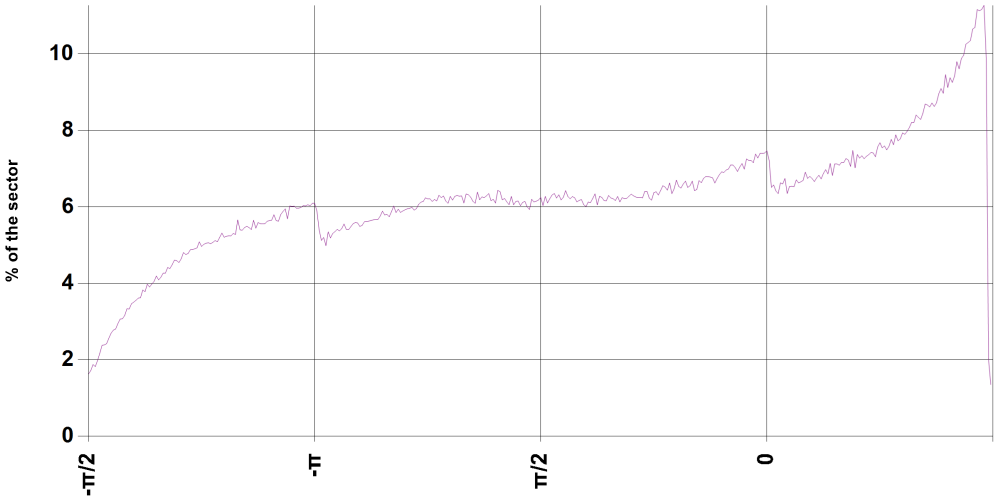


Figure 26: Spectrum of prime integers

Numerical Observation 5. *The density of prime integers, determined by the dynamics of the Square Rotor-Router, globally increases starting from $-\pi/2 + \delta\theta$, over an interval of 2π, following the clockwise direction.*

2.4 Study of Digit Positions

2.4.1 Objective of the Filter

We write

$$n = \sum_{k \geq 0} a_k \, 10^k, \quad a_k \in \{0, \ldots, 9\}.$$

The digit of rank k (position 10^k) is a_k. We study the distribution of a_k (for a fixed k) when $n \leq N$.

The clusters of integers n that satisfy $a_3 = 0$ are shown in white in the figures below.

2.4.2 Position $a_2 = 0$

Figure 27, which illustrates the case $a_2 = 0$ (that is, when the hundreds digit is zero), was generated with $N = 780 \times 10^3$ vertices. The zoom shown in Figure 28 corresponds to a cluster of integers located to the left of the center of the figure.

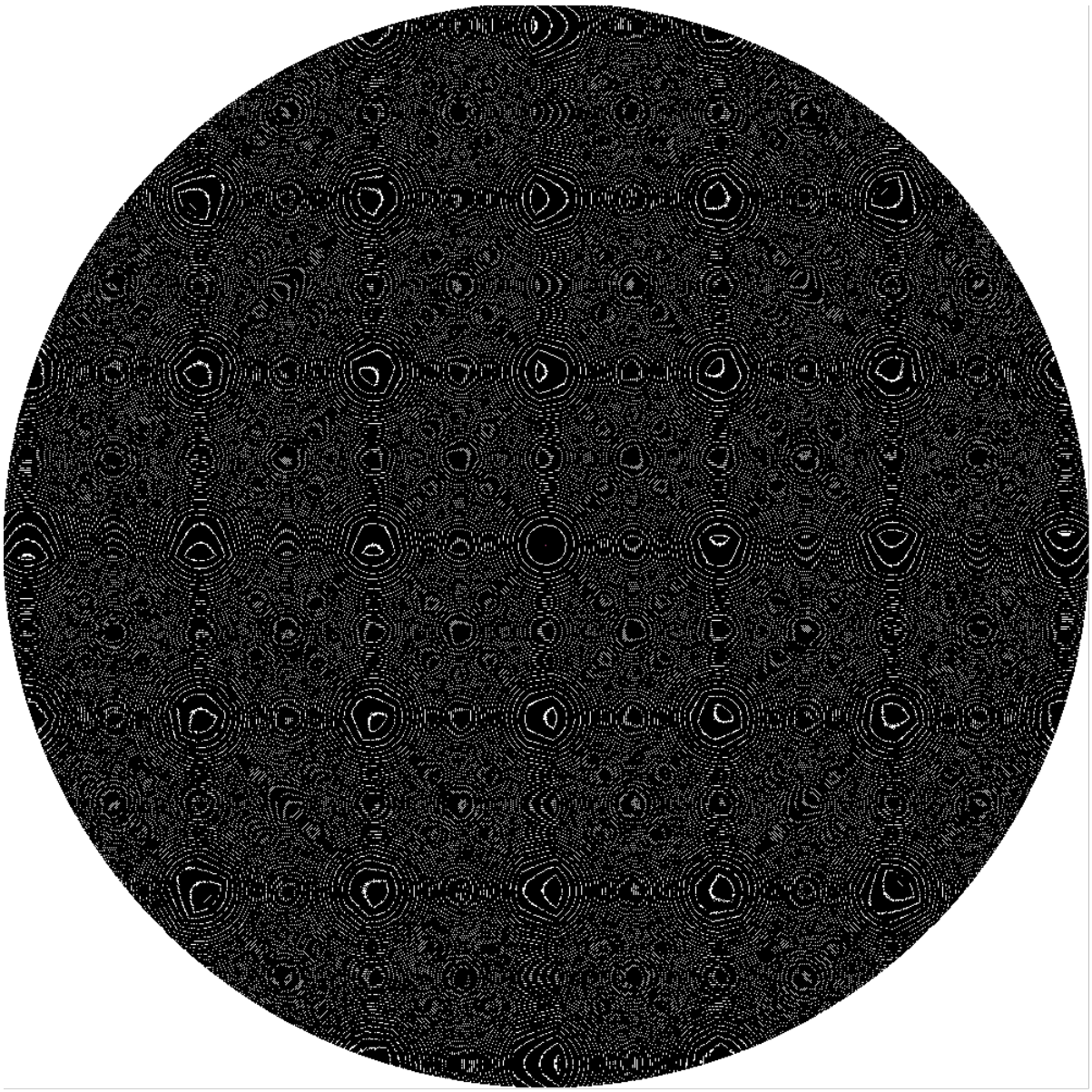


Figure 27: $a_2 = 0$ with $N = 780 \cdot 10^3$

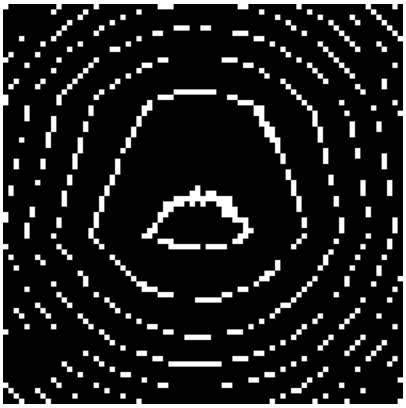


Figure 28: Zoom: $a_2 = 0$

Numerical Observation 6. *The integers n that satisfy $a_2 = 0$, determined by the dynamics of the Square Rotor-Router, are arranged in patterns reminiscent of fractals.*

2.4.3 Position $a_3 = 0$

Figure 29, for $a_3 = 0$ (the thousands digit is 0), was generated for $N = 28 \cdot 10^6$ vertex. The zooms in Figures 30 and 31 correspond to the cluster of integers below and to the left of the center of figure.

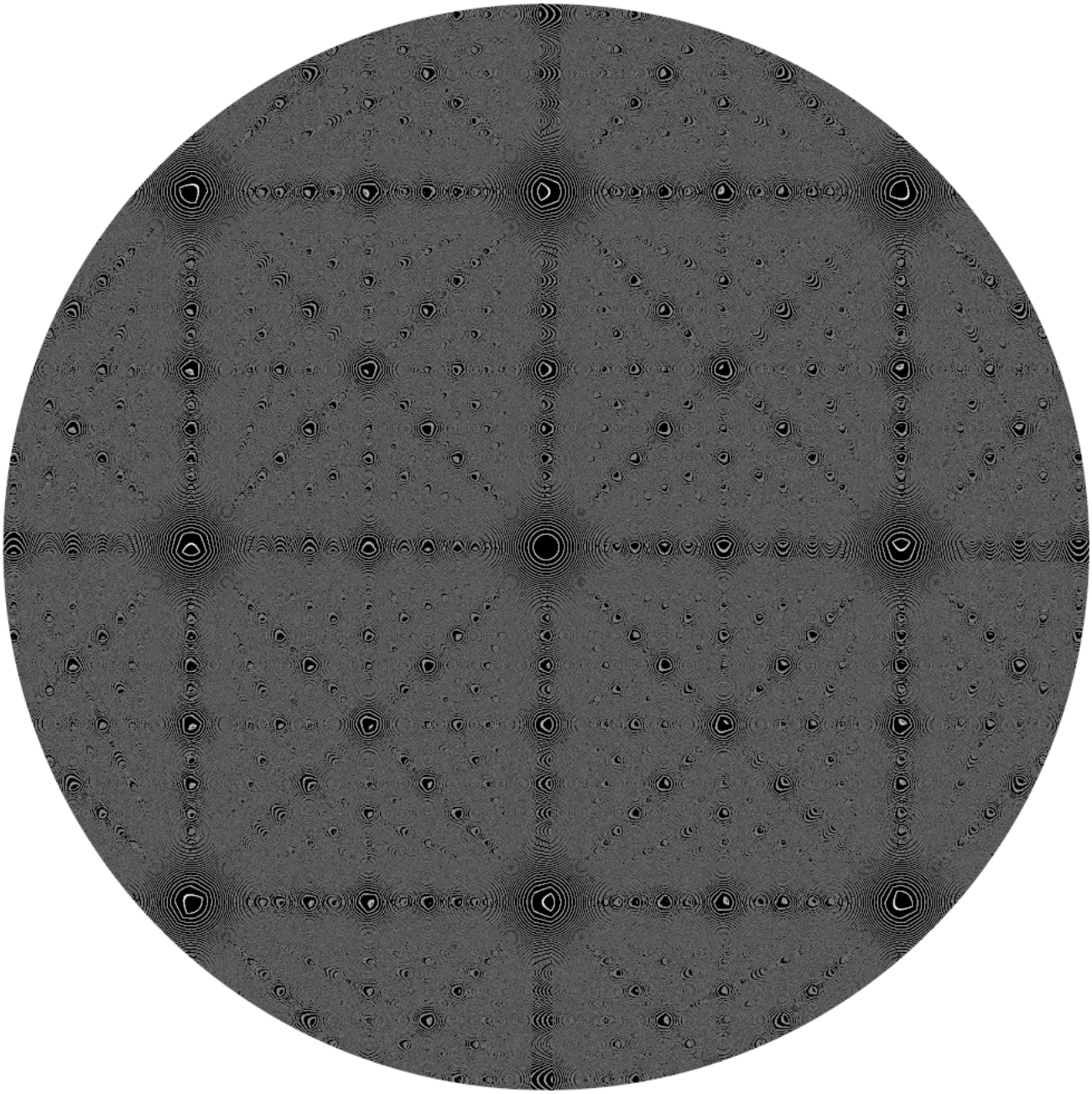


Figure 29: $a_3 = 0$ with $N = 28 \cdot 10^6$

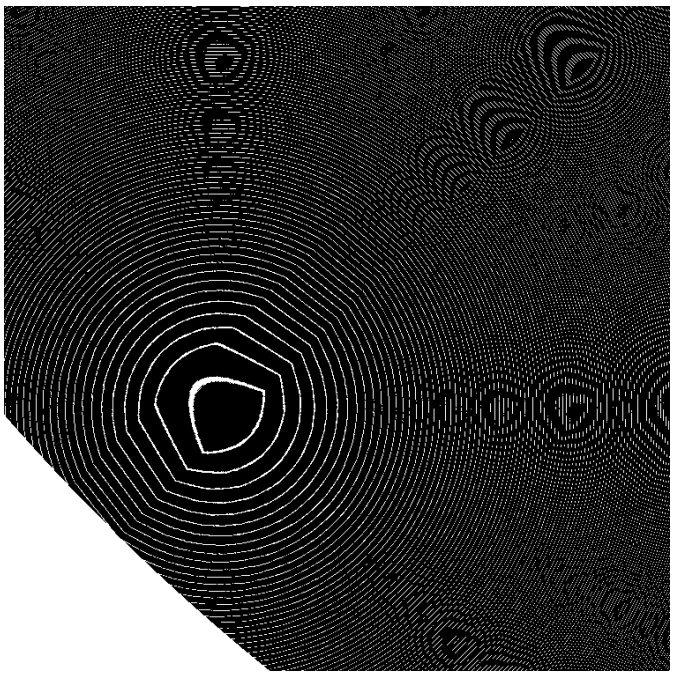


Figure 30: Zoom: $a_3 = 0$

Numerical Observation 7. *The integers n that satisfy $a_3 = 0$, determined by the dynamics of the Square Rotor-Router, are arranged in patterns reminiscent of fractals.*



Figure 31: Zoom: $a_3 = 0$

16520553	16510277	16500064								
16530645	16520381	16510149	16500036							
16540788	16530537	16520359								
16550942	16540778	16530188	16520119							
	16550576	16540537	16530124							
16570956	16560945	16550572	16540220							
	16570860	16560720	16550348	16540136						
		16570709	16560610	16550162						
		16580999	16570488	16560237	16550020					
		16590949	16580652	16570386	16560116					
			16590845	16580557	16570300					
			16600930	16590620	16580296	16570149				
				16600708	16590609	16580292				
				16610948	16600680	16590284				
					16610693	16600273	16590100			16550012
					16620749	16610560	16600232	16590140	16580288	16570385	16560304
					16630956	16620609	16610513	16600656	16590844	16580704	16570697
										16580976

Figure 32: The digits corresponding to the vertex framed above (12x18)

An image processing of the image makes it possible to highlight the fractal nature of the Rotor-Router for $a_3 = 0$.

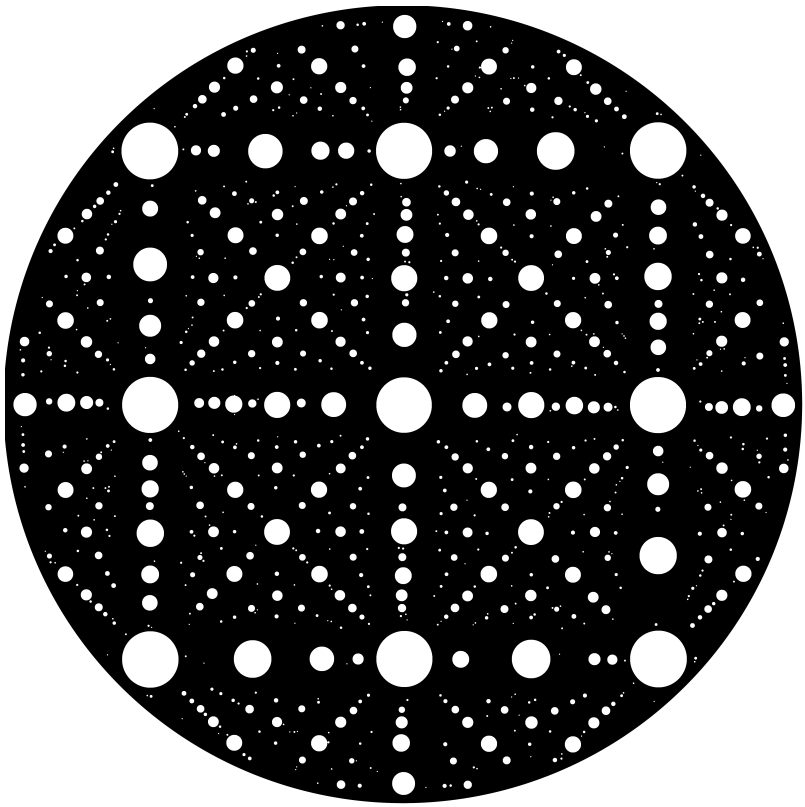


Figure 33: Image processing of the Rotor-Router with $a_3 = 0$

2.4.4 Generalization

The overall shape of the figure does not depend on a_k; for example, for $a_3 = 0 \ldots 9$, the positions of the clusters in the figure remain stable.

Numerical Observation 8. *For a given k, varying $a_k \in \{0, \ldots, 9\}$ does not change the overall shape of the structure. The patterns depend on k. They are not visible for $k = 0$ and $k = 1$. The examples above correspond to $k = 2$ and $k = 3$. For values of $k > 3$, the patterns would become more visible for $N > 50 \cdot 10^6$.*

3 Hexagonal Rotor-Router

3.1 Characteristics

Its characteristics are similar to those of the Square Rotor-Router, with the following differences:

- The number of neighbors of each vertex in this Rotor-Router is 6.
- The vertex are traversed in the order:
  - L for Left
  - LU for Left Up
  - RU for Right Up
  - R for Right
  - RD for Right Down
  - LD for Left Down

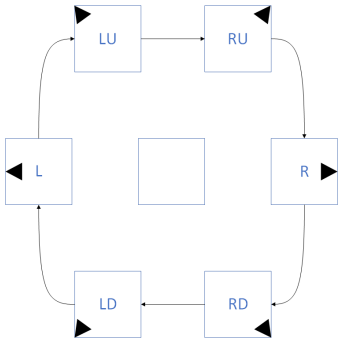


Figure 34: Direction of rotation of the Hexagonal Rotor-Router

- The colors associated with the orientations are as follows:

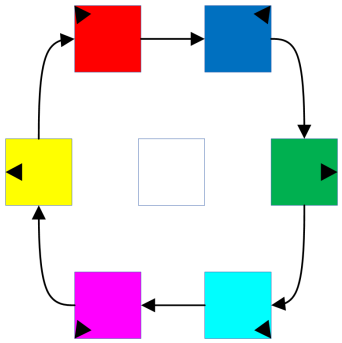


Figure 35: One color per direction

Remark. The display algorithm for Rotor-Router uses the pixel as a vertex. In the case of the Hexagonal Rotor-Router, the distance separating two vertex (the centers of two pixels) is larger vertically than horizontally. The resulting patterns therefore take the form of a vertical ellipse. The ratio between width and height is $\frac{2}{\sqrt{5}} \approx 0.8944$. In the remainder of this study, the images of the Hexagonal Rotor-Router will be corrected by this ratio in order to obtain a circle.

Figure 36: Regular hexagon vs. shape of the hexagon obtained by the algorithm

The following figure was generated for integers $N = 16 \cdot 10^6$.

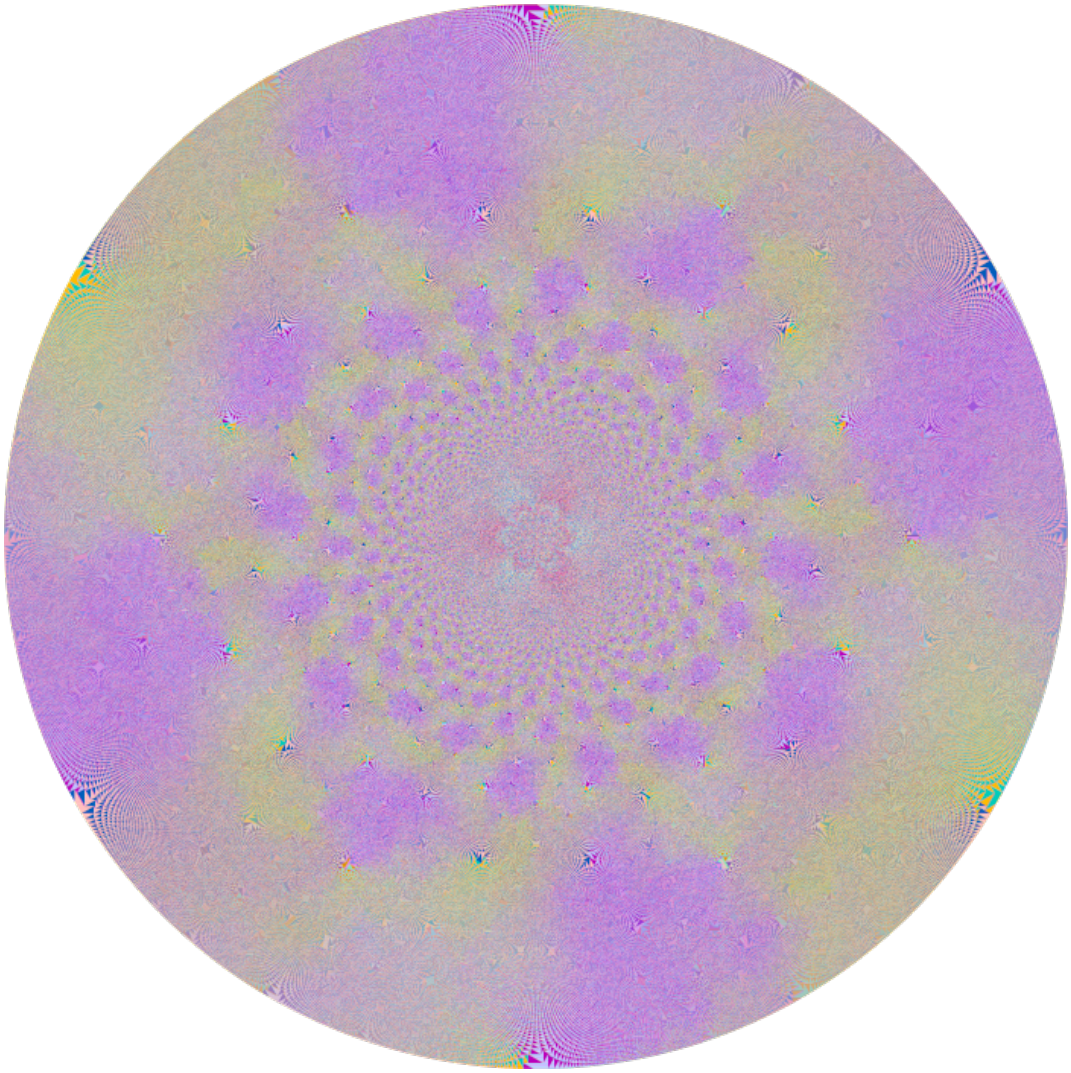


Figure 37: Hexagonal Rotor-Router pattern related to the orientation of each cell.

4 Observed Results on the Hexagonal Rotor-Router

The results observed with the Hexagonal Rotor-Router are, in part, similar to those obtained with the Square Rotor-Router. We will not revisit all the results of section 2.

The Square Rotor-Router partitions the disk into 4 parts, while this one seems to partition it into 6 parts. Starting from $-\pi/2$: $-\pi/2 \quad -2\pi/3 \quad 2\pi/3 \quad \pi/2 \quad \pi/3 \quad -\pi/3$.

The spectral curves are inverted, and there is also a shift $\delta\theta$ with respect to $-\pi/2$.

The structure of the clusters obtained by filtering according to the position of the digits remains fractal. The resulting fractal shape is a hexagon.

4.1 Divisibility Filter

4.1.1 Filter $n = 2k$

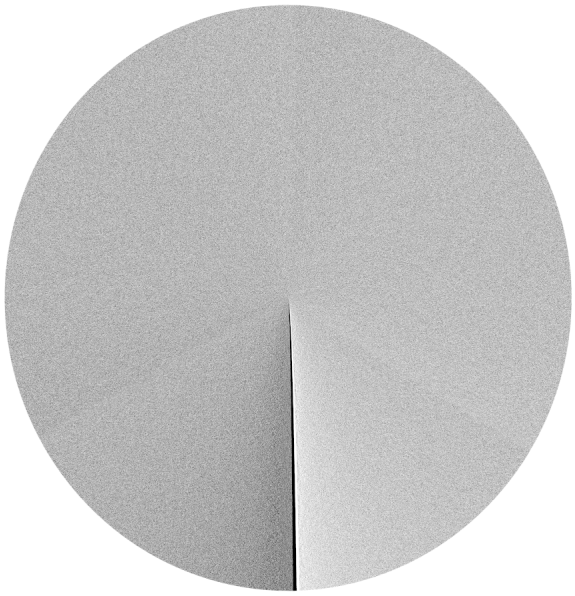


Figure 38: Integers of the form $n = 2k$

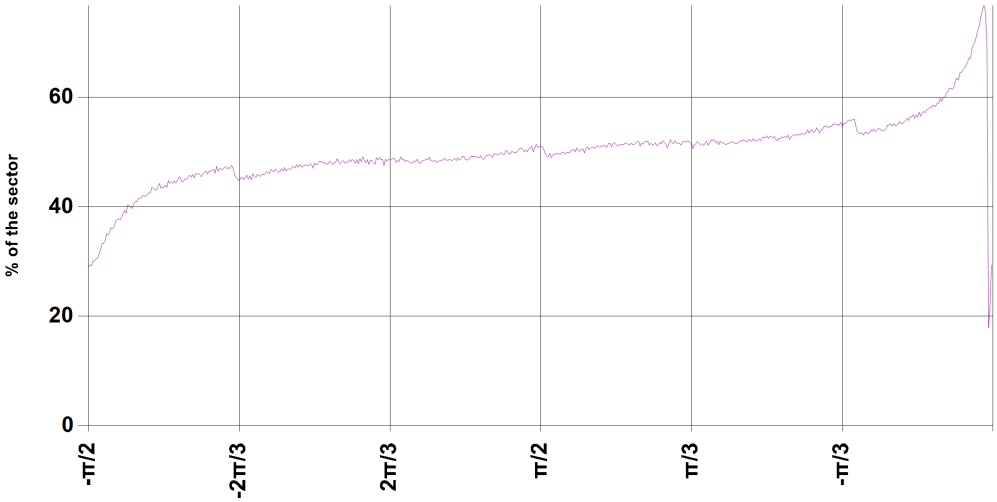


Figure 39: Spectrum of integers $n = 2k$

Numerical Observation 9. *The density of integers of the form $n = 2k$ with $k \in \mathbb{N}^*$, determined by the dynamics of the Hexagonal Rotor-Router, globally increases starting from $-\pi/2 + \delta\theta$, over an interval of 2π, following the clockwise direction.*

4.1.2 Filter $n = 3k$

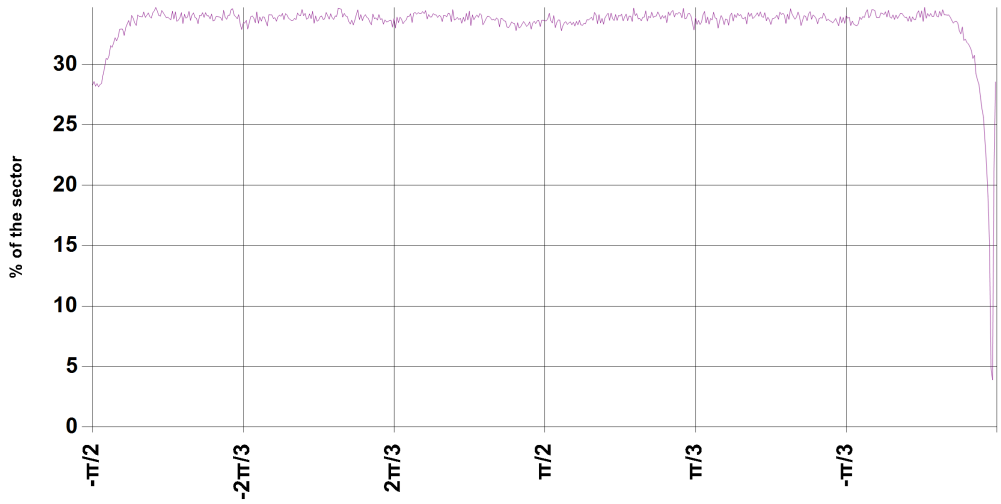


Figure 40: Spectrum of integers $n = 3k$

4.2 Congruence

Let us consider integers n of the form $n \equiv 4 \pmod 6$, and then $n \equiv 5 \pmod 6$.

Appendix B presents the spectra of some congruence classes $b \pmod a$, with $\gcd(b, a) = 1$.

4.2.1 $n \equiv 4 \pmod 6$

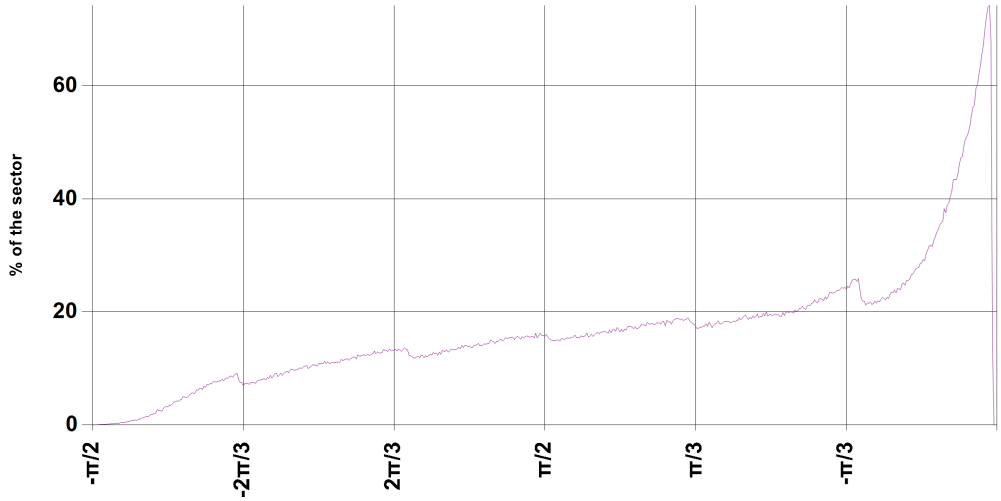


Figure 41: Spectrum of integers $n \equiv 4 \pmod 6$

4.2.2 $n \equiv 5 \pmod 6$

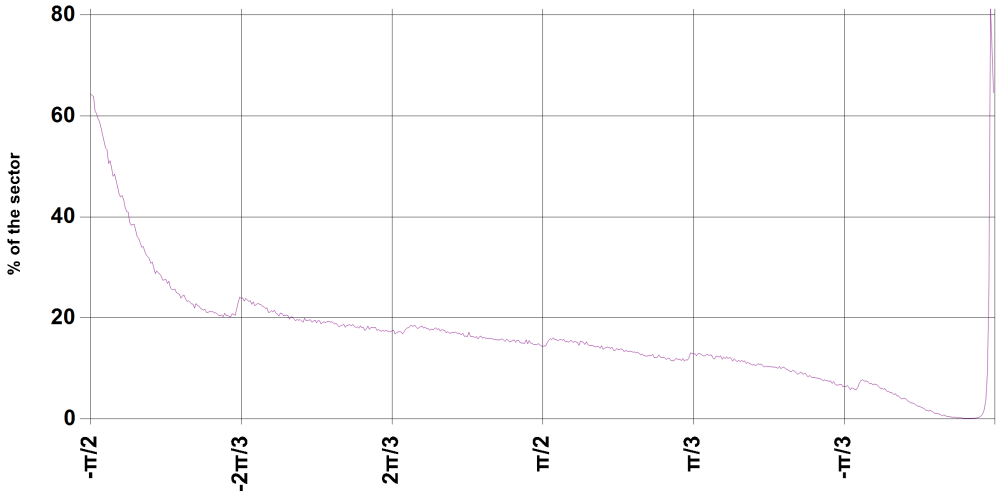


Figure 42: Spectrum of integers $n \equiv 5 \pmod 6$

4.3 Prime Integers

If we retain only the prime numbers:

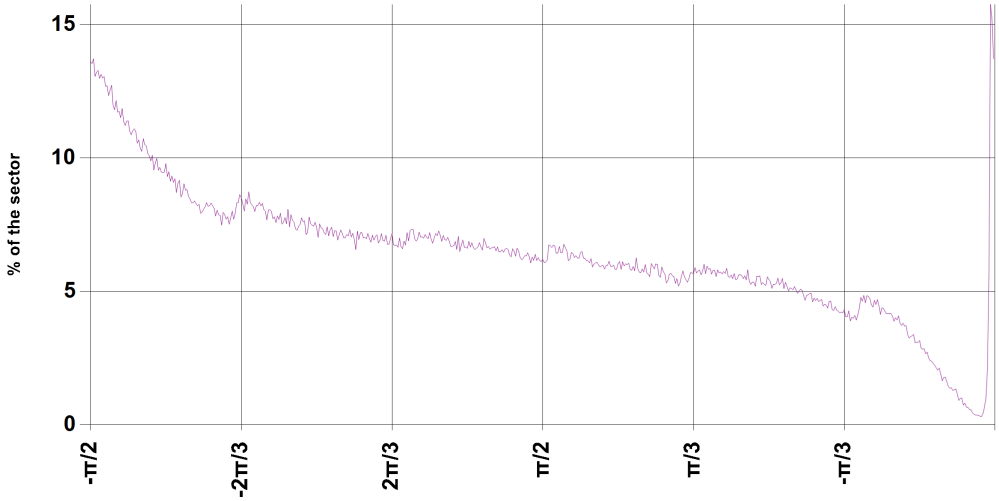


Figure 43: Spectrum of prime integers

4.4 Study of digit positions

4.4.1 Position $a_2 = 0$

Figure 44, which illustrates the case $a_2 = 0$ (that is, when the hundreds digit is zero), was generated with $N = 16{\cdot}10^6$ vertices.

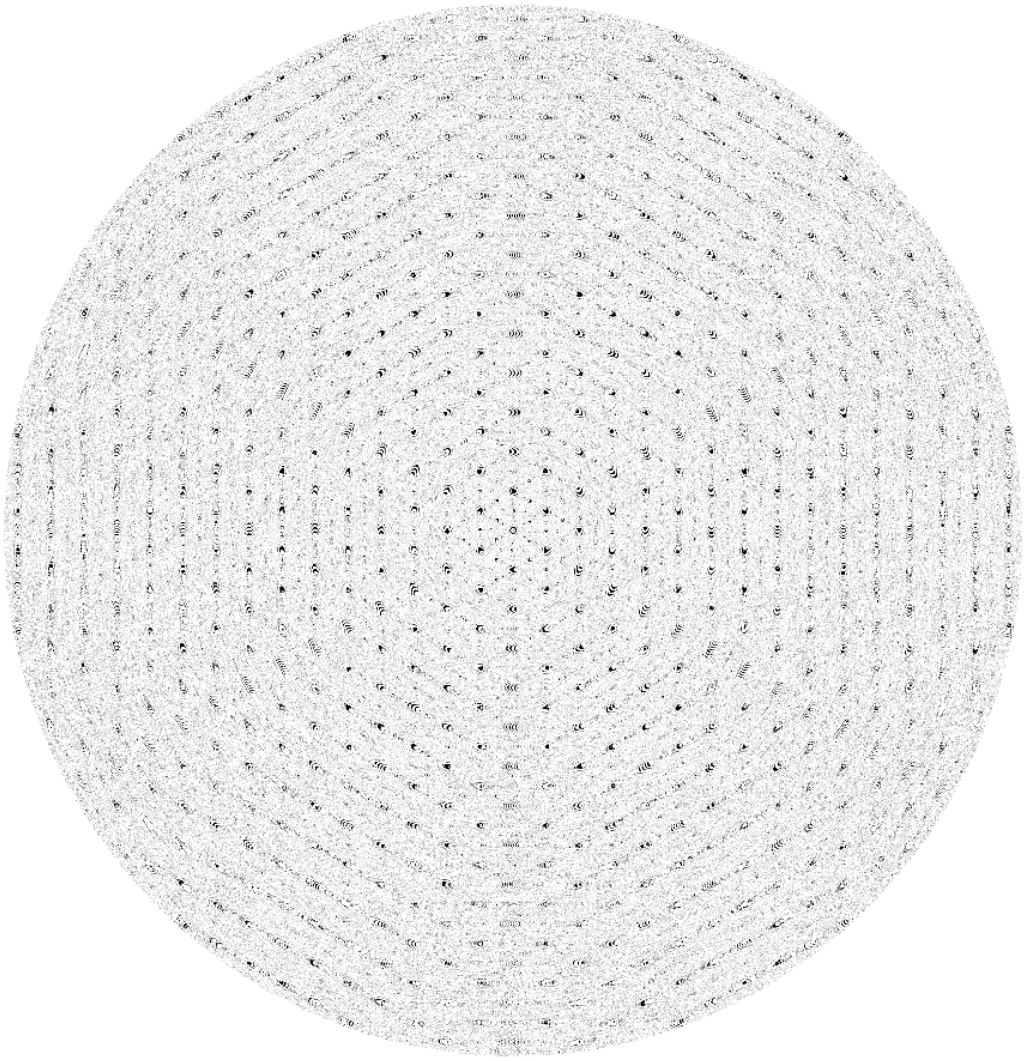


Figure 44: $a_2 = 0$ with $N = 16 \cdot 10^6$

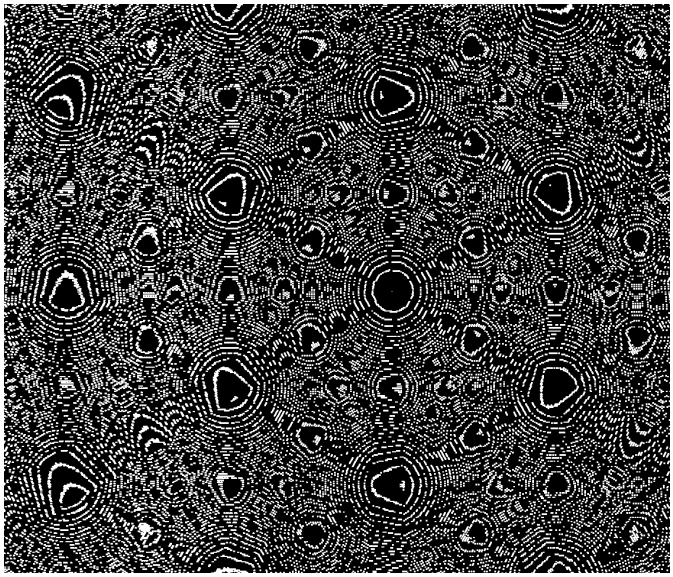


Figure 45: Zoom at the center for $a_2 = 0$

Numerical Observation 10. *The integers n satisfying $a_2 = 0$, determined by the Hexagonal Rotor-Router dynamics, arrange themselves into patterns reminiscent of fractal structures.*

4.4.2 Position $a_3 = 0$

Figure 46, corresponding to the case $a_3 = 0$ (that is, when the thousands digit is zero), was generated with $N = 28 \cdot 10^6$ vertices.

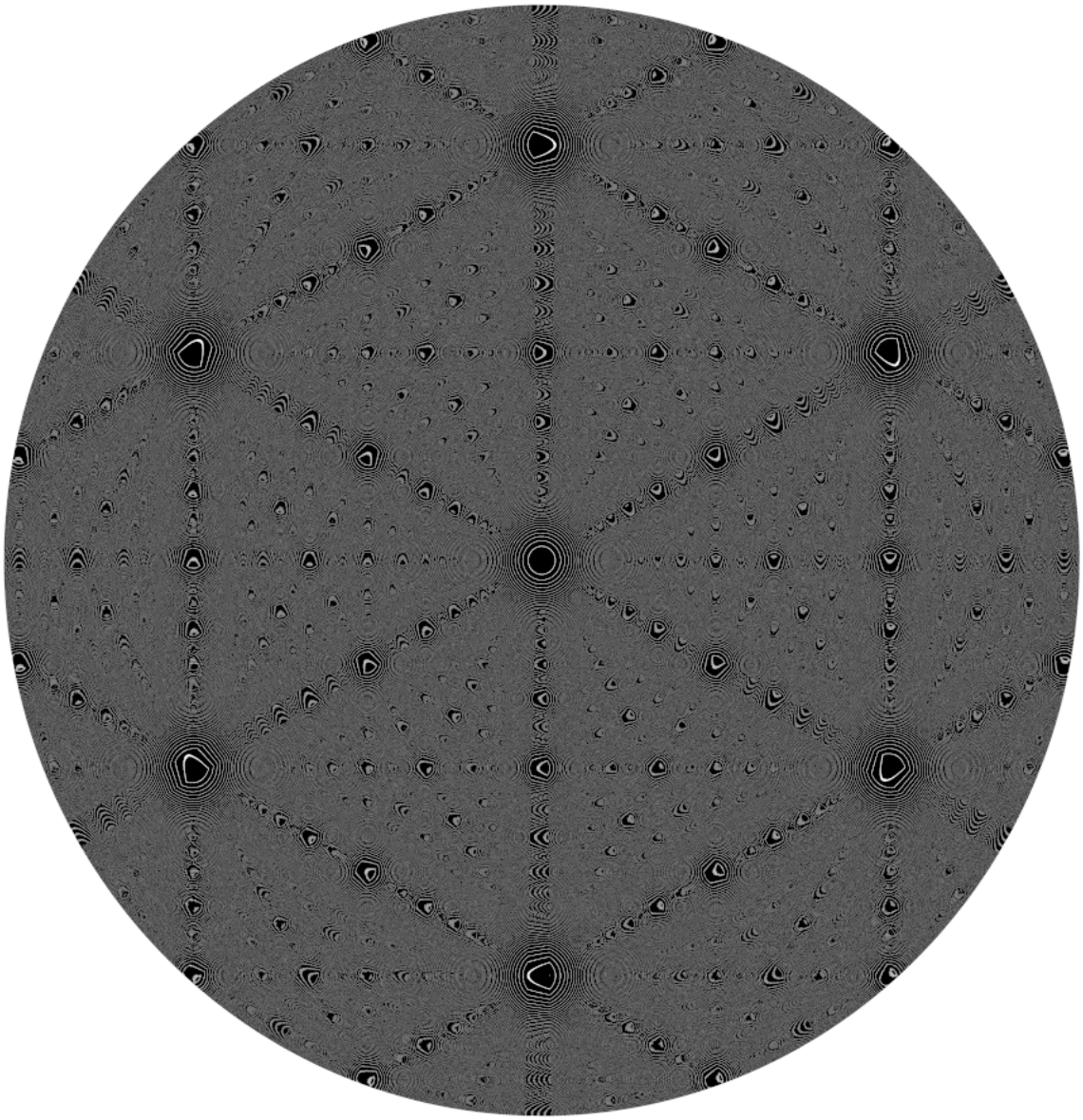


Figure 46: $a_3 = 0$ with $N = 28 \cdot 10^6$

Numerical Observation 11. *The integers n satisfying $a_3 = 0$, determined by the Hexagonal Rotor-Router dynamics, arrange themselves into patterns reminiscent of fractal structures.*

Image processing highlights the fractal nature of the Hexagonal Rotor-Router for $a_3 = 0$.

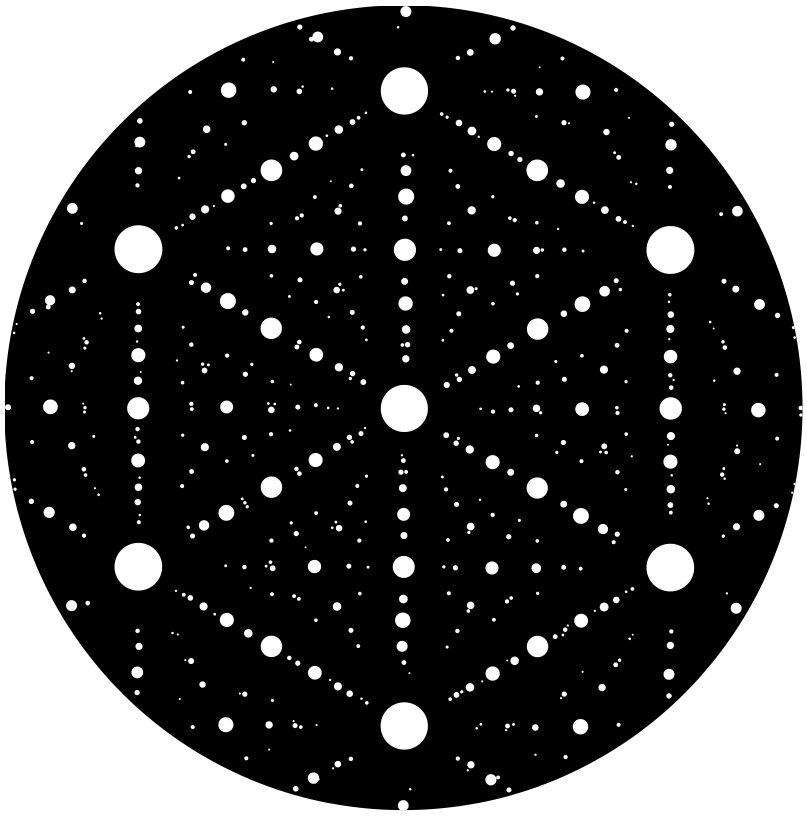


Figure 47: Image processing of the Hexagonal Rotor-Router for $a_3 = 0$

4.4.3 Generalization

Numerical Observation 12. *For a given k, varying $a_k \in \{0, \dots, 9\}$ does not change the global shape of the structure. However, the patterns depend on k: they are not visible for $k = 0$ and $k = 1$. The above examples correspond to $k = 2$ and $k = 3$. For values $k > 3$, the patterns become more visible when $N > 50 \cdot 10^6$.*

5 Analysis of particle trajectories

5.1 Numerical analysis

This analysis is carried out on the Square Rotor-Router.

subsection 1.3 describes the behavior of particles that follow shorter or longer trajectories before reaching the exterior of the disk. The study of these paths has revealed an interesting feature.

This analysis required collecting and recording the trajectories of $2 \cdot 10^5$ particles, with $n \in [1; 2 \cdot 10^5]$. When n is small, the trajectory length remains short, but it increases significantly as n grows.

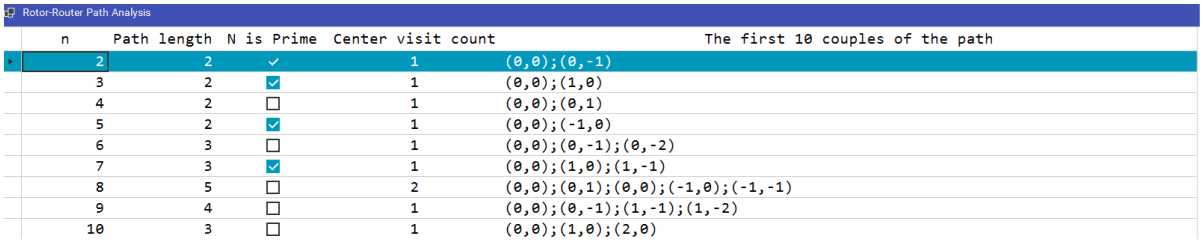
As an example, for $n = 184{,}132$, the trajectory consists of 1,015,595 steps.

Many studies on Rotor-Router models focus on the probability that a particle returns to the origin. In our study, we focus on particles that never pass through the origin again after being deposited there.

The collected data were stored in a file (with a size exceeding 50 GB) and consist of:

- Particle index.
- Path length (number of steps required before reaching the exterior of the disk).
- Path taken (the list of (x, y) pairs describing the trajectory).

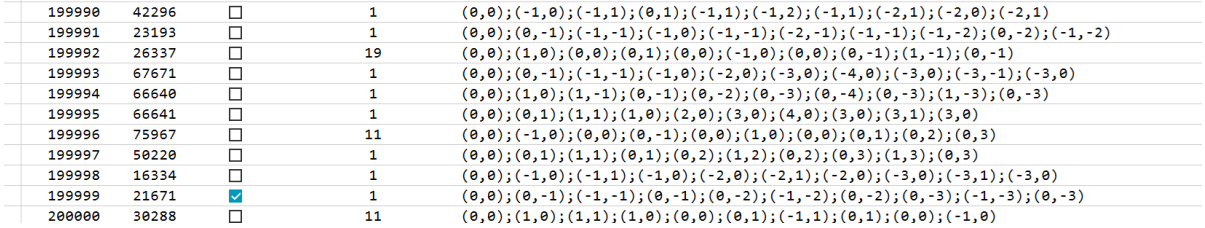
The data are summarized in a representative table:



Rotor-Router Path Analysis

n	Path length	N is Prime	Center visit count	The first 10 couples of the path
2	2	✓	1	(0,0);(0,-1)
3	2	☑	1	(0,0);(1,0)
4	2	☐	1	(0,0);(0,1)
5	2	☑	1	(0,0);(-1,0)
6	3	☐	1	(0,0);(0,-1);(0,-2)
7	3	☑	1	(0,0);(1,0);(1,-1)
8	5	☐	2	(0,0);(0,1);(0,0);(-1,0);(-1,-1)
9	4	☐	1	(0,0);(0,-1);(1,-1);(1,-2)
10	3	☐	1	(0,0);(1,0);(2,0)

Figure 48: The first particles



199990	42296	☐	1	(0,0);(-1,0);(-1,1);(0,1);(-1,1);(-1,2);(-1,1);(-2,1);(-2,0);(-2,1)
199991	23193	☐	1	(0,0);(0,-1);(-1,-1);(-1,0);(-1,-1);(-2,-1);(-1,-1);(-1,-2);(0,-2);(-1,-2)
199992	26337	☐	19	(0,0);(1,0);(0,0);(0,1);(0,0);(-1,0);(0,0);(0,-1);(1,-1);(0,-1)
199993	67671	☐	1	(0,0);(0,-1);(-1,-1);(-1,0);(-2,0);(-3,0);(-4,0);(-3,0);(-3,-1);(-3,0)
199994	66640	☐	1	(0,0);(1,0);(1,-1);(0,-1);(0,-2);(0,-3);(0,-4);(0,-3);(1,-3);(0,-3)
199995	66641	☐	1	(0,0);(0,1);(1,1);(1,0);(2,0);(3,0);(4,0);(3,0);(3,1);(3,0)
199996	75967	☐	11	(0,0);(-1,0);(0,0);(0,-1);(0,0);(1,0);(0,0);(0,1);(0,2);(0,3)
199997	50220	☐	1	(0,0);(0,1);(1,1);(0,1);(0,2);(1,2);(0,2);(0,3);(1,3);(0,3)
199998	16334	☐	1	(0,0);(-1,0);(-1,1);(-1,0);(-2,0);(-2,1);(-2,0);(-3,0);(-3,1);(-3,0)
199999	21671	☑	1	(0,0);(0,-1);(-1,-1);(0,-1);(0,-2);(-1,-2);(0,-2);(0,-3);(-1,-3);(0,-3)
200000	30288	☐	11	(0,0);(1,0);(1,1);(1,0);(0,0);(0,1);(-1,1);(0,1);(0,0);(-1,0)

Figure 49: The last particles

- Column 1: Particle index n.
- Column 2: Path length.
- Column 3: Indicator "n prime or not".
- Column 4: Number of visits to the origin.
- Column 5: The first 10 steps.

Let $S_N = \{\, n \le N : \text{particle } n \text{ never returns to the origin} \,\}$, and denote its cardinality by $|S_N|$.
Let $Q_N = \{\, n \le N : n \text{ is not prime} \,\}$. We denote its cardinality by $|Q_N|$.
Let $P_N = \{\, n \le N : n \text{ is prime} \,\}$. We denote its cardinality by $|P_N|$.
Let $\overline{L}_N = \frac{1}{N}\sum_{n \le N} L(n)$ denote the average trajectory length of the first N particles.
Let $M_N = \{\, n \le N : 4 \nmid n \,\}$. We denote its cardinality by $|M_N|$.

We first group the data according to the criterion "prime or not".

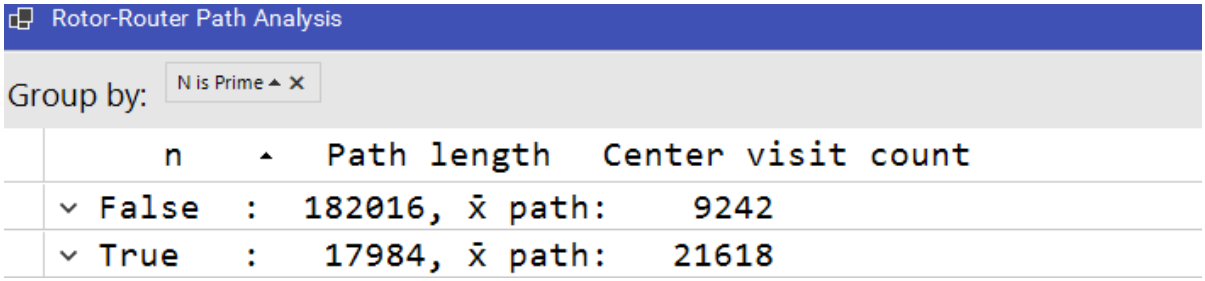


Figure 50: Grouping according to the criterion "prime or not"

Figure 50 separates the $2 \cdot 10^5$ particles, indexed from 1 to $2 \cdot 10^5$, into two groups:

- $|Q_N| = 182{,}016$, $\overline{L}_N = 9{,}142$.
- $|P_N| = 17{,}984$, $\overline{L}_N = 21{,}618$.
- $|Q_N| + |P_N| = 200{,}000$.

We then group the data according to a second criterion: "the number of visits to the origin".

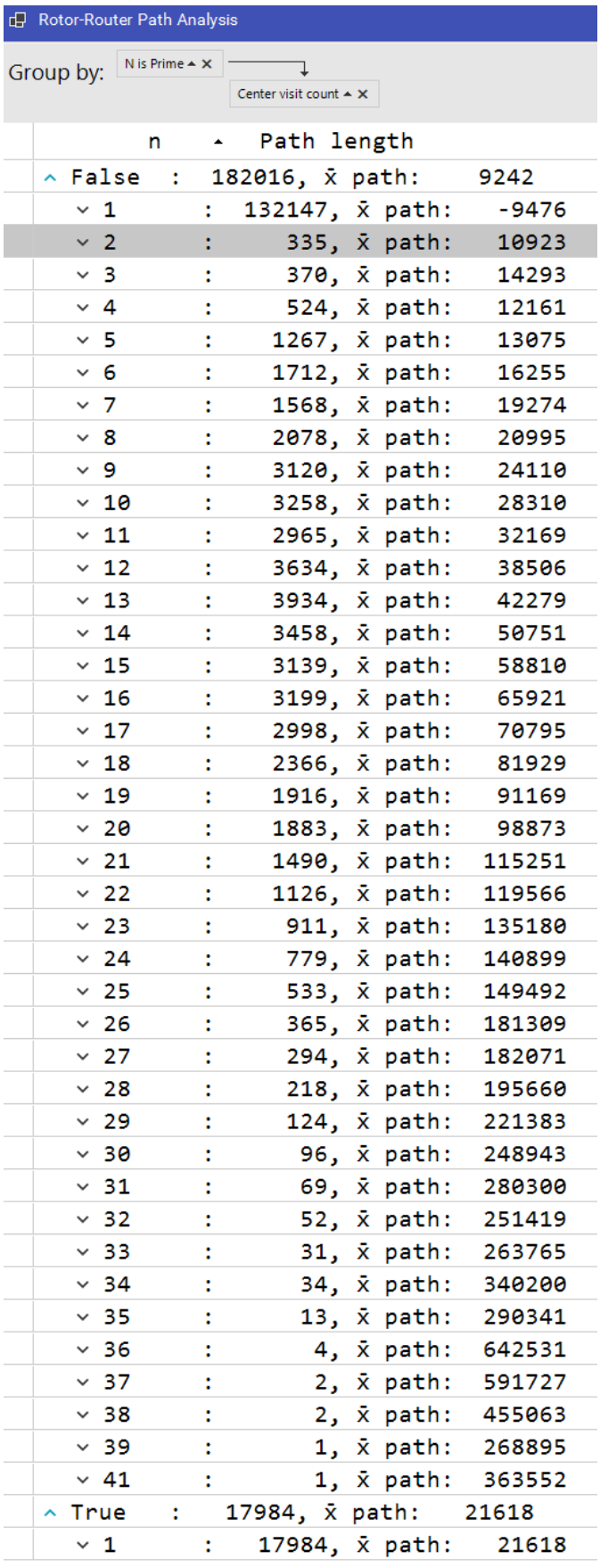


Figure 51: Grouping according to the criteria “prime” and “number of visits to the origin”

We observe that, for prime indices, there is only one visit to the origin (the initial deposition). This is not the case for non-prime indices.

Numerical Observation 13. *The analysis of particle trajectories in the Square Rotor-Router shows that particles with prime indices never return to the origin during their paths.*

5.2 Relation to the non-multiples of 4.

There appears to be a connection between the particles that never return to the origin and the integers that are not multiples of 4, i.e. those of the form $n \neq 4k$.

Figure 18 illustrates the spectrum of integers of the form $n = 4k$ for $N = 28 \cdot 10^6$.

Figure 52 shows the spectrum of integers of the form $n \neq 4k$ for $N = 2 \cdot 10^5$. $|M_N| = 150{,}000$.

Figure 53 depicts the spectrum of the particles that never return to the origin for $N = 2 \cdot 10^5$. $|S_N| = 150{,}131$.

Numerical Observation 14. *The spectrum of indices of the trajectories followed by the particles of the Square Rotor-Router that never return to the origin is very similar to that of the integers of the form $n \neq 4k$.*

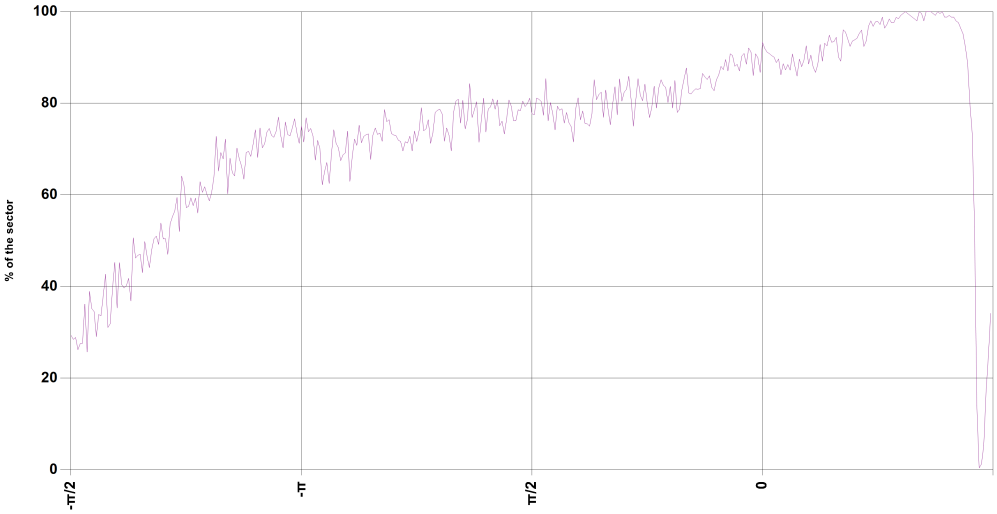


Figure 52: Integers of the form $n \neq 4k$ for $N = 2 \cdot 10^5$.

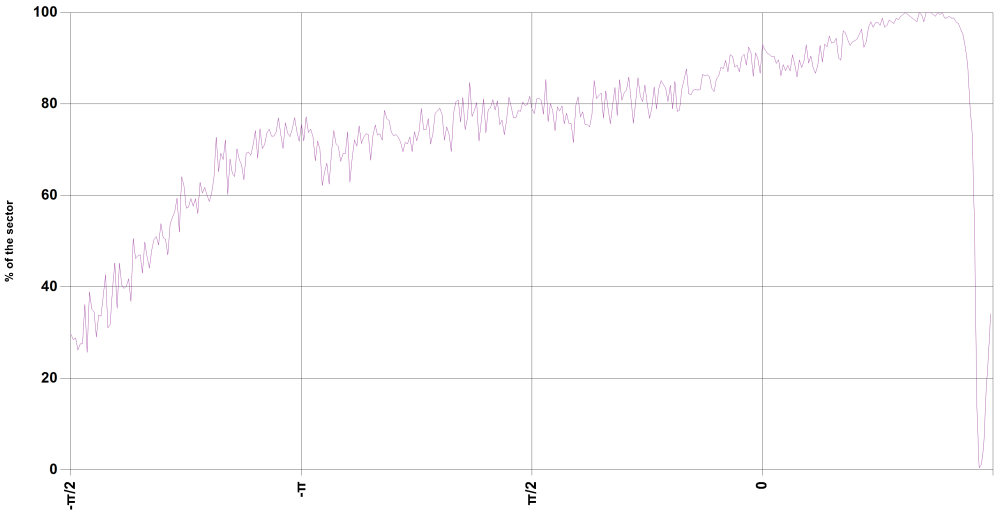


Figure 53: Particles that never return to the origin for $N = 2 \cdot 10^5$.

- For $N = 10^5$, $|M_N| = 75{,}000$ and $|S_N| = 75{,}099$.
- For $N = 2 \cdot 10^5$, $|M_N| = 150{,}000$ and $|S_N| = 150{,}131$.

Numerical Observation 15. *The relative difference between $|M_N|$ and $|S_N|$ appears to decrease as N increases.*

6 Conclusions and Conjectures

6.1 Conclusions

What conclusions can be drawn from this numerical study?

Conclusion 1. *The distribution of the integer associated with the particle indices in a Rotor-Router aggregation model is not random. The application of numerical or functional filters reveals surprising biases.*

Conclusion 2. *These biases are not limited to the Square Rotor-Router model: they also appear in the Hexagonal Rotor-Router model.*

Conclusion 3. *With the conventions adopted in this study (rotation direction, ordering), integers belonging to certain congruence classes appear to concentrate within one quadrant of the circle.*

Conclusion 4. *The concentrations of filtered integers depend on the order of the Rotor-Router model: four zones for the Square Rotor-Router model and six zones for the Hexagonal Rotor-Router model.*

Conclusion 5. *Prime numbers exhibit similar concentrations.*

Conclusion 6. *The position of digits within integers generates fractal-like patterns. These patterns depend on the order of the model (Square Rotor-Router or Hexagonal Rotor-Router) and retain the same overall structure as the digit of rank a_k varies.*

Conclusion 7. *Integers of the form $n \neq 4k$ (and therefore the prime numbers) appear to be connected with the trajectories of the particles, and in particular with their returns to the origin.*

Conclusion 8. *There appears to be a connection between the number of particles that return to the origin and the set of integers not divisible by 4.*

6.2 Conjectures

Preliminary note. Rotor-Router aggregation models are tools for visualizing the distribution of integers.

Let us venture a few conjectures:

Conjecture 1. *Rotor-Router aggregation models reveal concentrations both of congruence classes and of prime numbers.*

Conjecture 2. *The position of digits in the indices of particles in Rotor-Router aggregation models, when projected onto the Rotor-Router plane, gives rise to fractal-like patterns.*

Conjecture 3. *In Rotor-Router aggregation models, deposited prime numbers follow trajectories that never return to the origin.*

Conjecture 4. *The relative difference between $|S_N|$ and $|M_N|$ tends to zero, that is*

$$\lim_{N\to\infty} \frac{|S_N| - |M_N|}{|M_N|} = 0.$$

and

$$\frac{N - |M_N|}{N} = \frac{\lfloor N/4 \rfloor}{N} \longrightarrow \tfrac{1}{4} \quad \text{as } N \to \infty.$$

6.3 Further Directions

6.3.1 Rotor-routers of different orders

This study was limited to the Square Rotor-Router and Hexagonal Rotor-Router models. An attempt with the triangular Rotor-Router model produced results very similar to those of the Hexagonal Rotor-Router. In contrast, the graphical representation of pentagonal Rotor-Router models is not straightforward, unless one considers them on a hyperbolic disk.

6.3.2 Performance improvements

For the Square Rotor-Router model, this study was limited to the deposition of $28 \cdot 10^6$ particles. The model size is constrained by computer performance.

Parallelization of the program in Appendix C did not yield the expected results: locking and concurrency constraints degraded performance.

In their article [6], Lionel Levine and Yuval Peres propose an algorithm to process particle emissions from the origin in parallel. This approach may warrant further investigation.

6.3.3 Other filters

The following filters were applied:

- The number of positive divisors of n:
$$d(n) = |\{\, k \in \mathbb{N} \mid k \mid n \,\}|\,.$$
- Goldbach decompositions, which consist of writing an even integer n as the sum of two prime numbers:
$$n = p + q, \qquad p, q \in \mathbb{P}.$$

These tests did not yield conclusive results. Other filters, however, might highlight new biases.

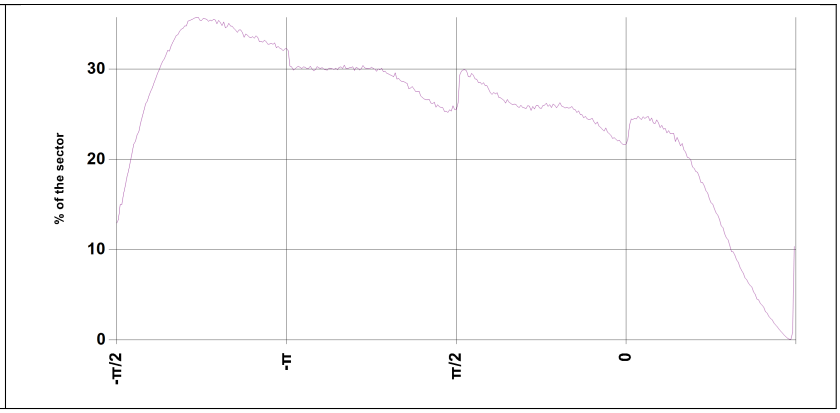
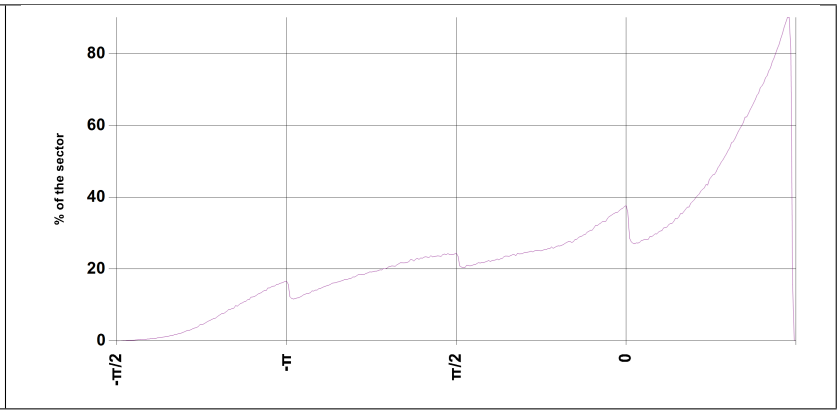
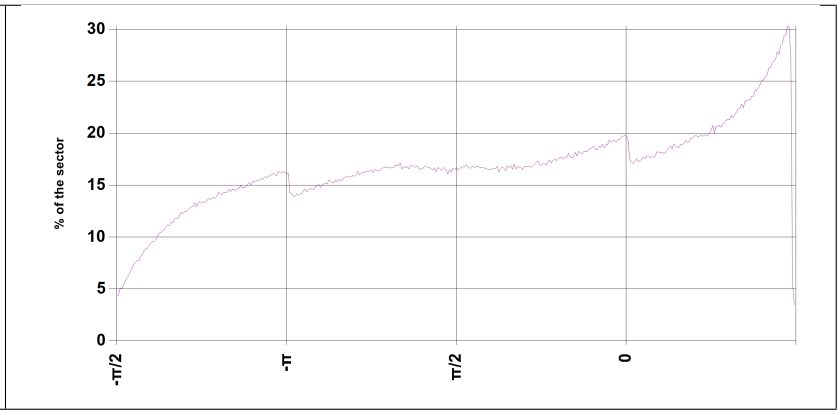
6.3.4 Fractal-like patterns

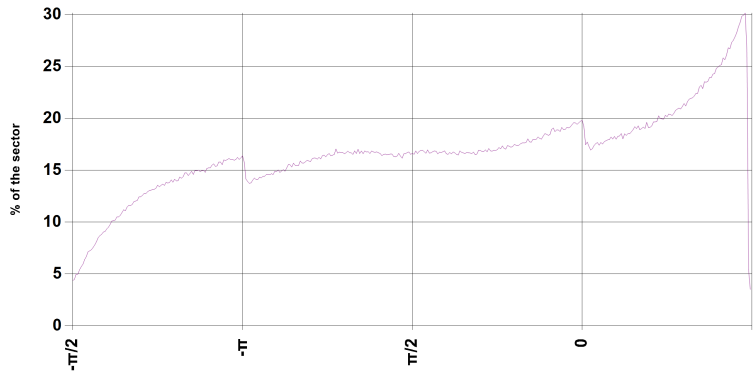
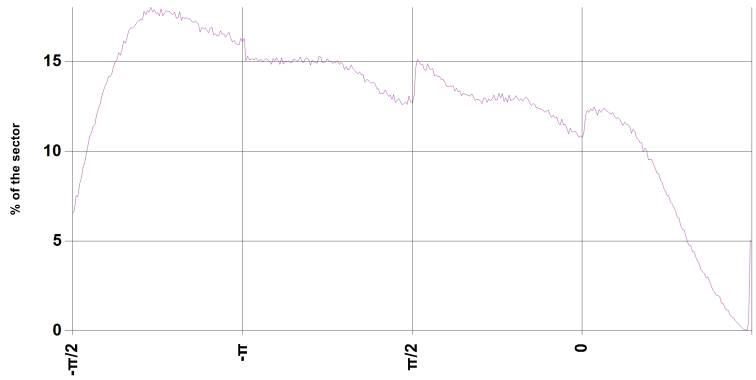
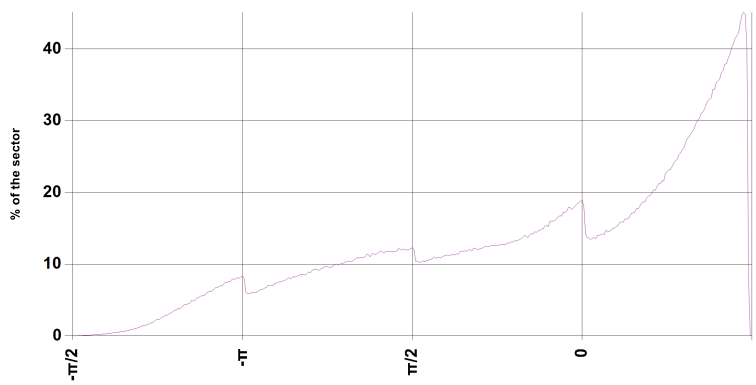
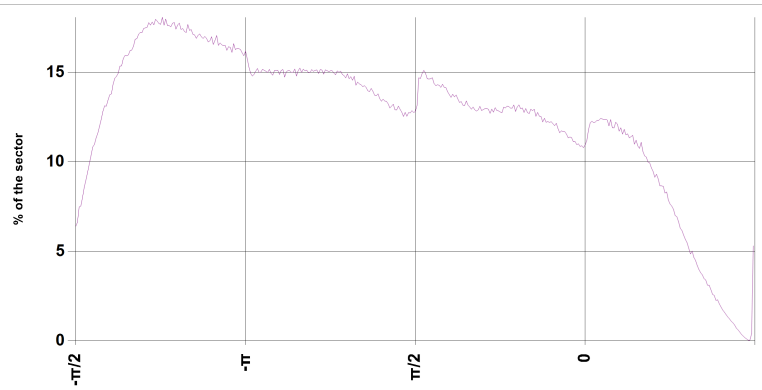
The connection between the position of a digit in the integer associated with the index of a particle in a Rotor-Router model appears non-trivial. The fractal patterns that emerge deserve deeper analysis.

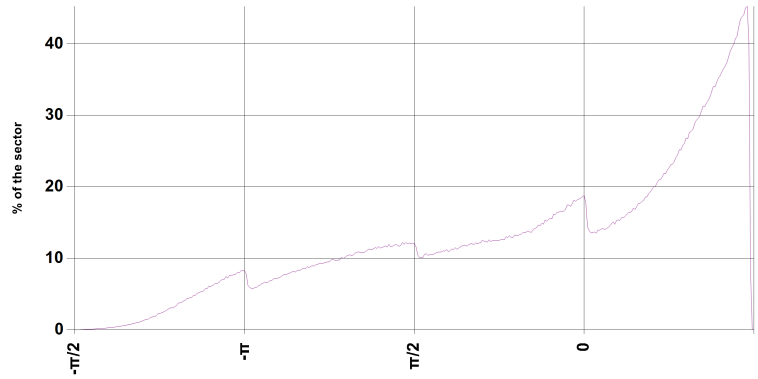
6.3.5 Particle paths and primality

Among all the vertices that host particles, the origin seems to play a special role. Indeed, prime-indexed particles never pass through the origin again during their trajectories. The study of these trajectories also deserves further analysis.

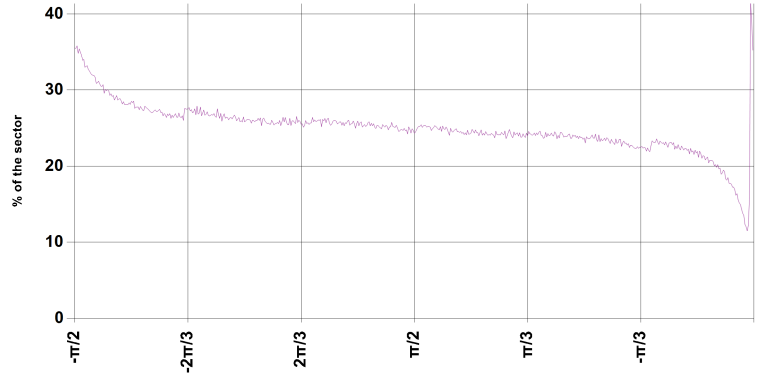
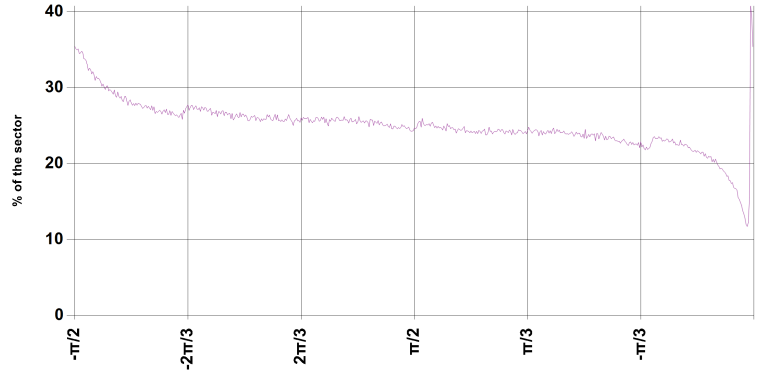
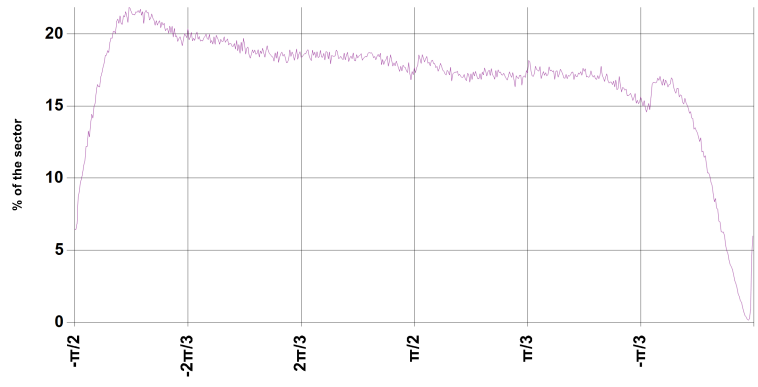
A Congruence Classes of the Square Rotor-Router

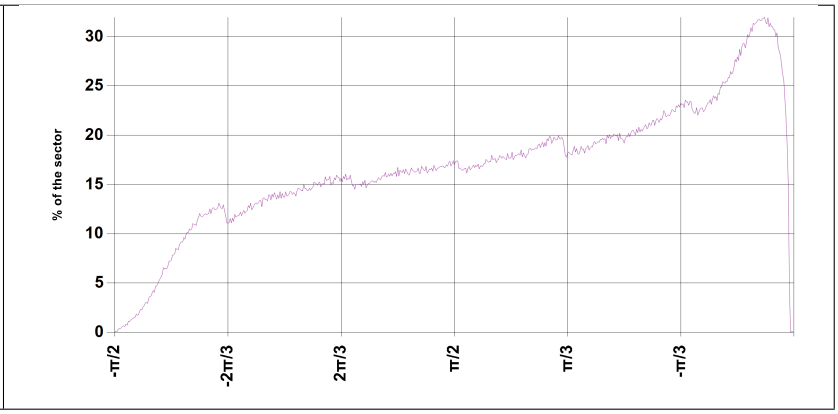
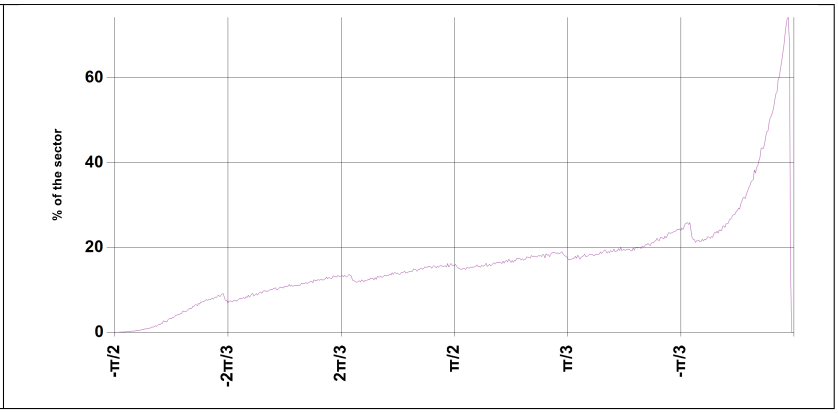
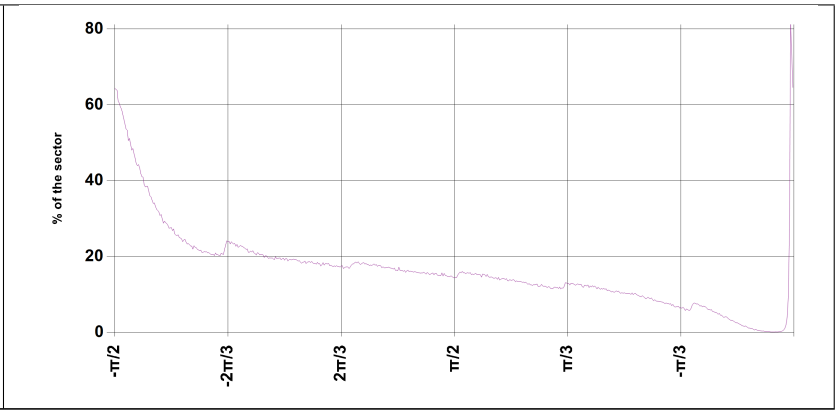
Congruence class $n \equiv 1 \pmod 4$	
Congruence class $n \equiv 3 \pmod 4$	
Congruence class $n \equiv 1 \pmod 6$	

Congruence class $n \equiv 5 \pmod 6$	
Congruence class $n \equiv 1 \pmod 8$	
Congruence class $n \equiv 3 \pmod 8$	
Congruence class $n \equiv 5 \pmod 8$	

Congruence class $n \equiv 7 \pmod 8$	

B Congruence Classes of the Hexagonal Rotor-Router

Congruence class $n \equiv 1 \pmod 4$	
Congruence class $n \equiv 3 \pmod 4$	
Congruence class $n \equiv 1 \pmod 6$	

Congruence class $n \equiv 3 \pmod 6$	
Congruence class $n \equiv 4 \pmod 6$	
Congruence class $n \equiv 5 \pmod 6$	

C Code for Chip Placement for Square Rotor-Router

```
double ProcessArray(uint32_t* __restrict arrayCell, int size, int currentStart, int count)
{
  if (count <= 0) return 0.0;
  const int center = size / 2;
  const int baseIndex = center * size + center;

  // Constants (defined elsewhere as constexpr)
  constexpr unsigned SHIFT = static_cast<unsigned>(shiftorientation); // 26
  constexpr uint32_t ORIENT_MASK = orientation_mask;    // 0xFu << SHIFT (using 0..3)
  constexpr uint32_t VALUE_MASK_ = VALUE_MASK;      // 26 bits
  constexpr uint32_t MASK_DISPLAY = MaskDisplay;    // bit 31 (must end at 0)
  constexpr uint32_t MASK_IS_SET = mask_is_set;   // bit 30 (must end at 1)

  // Hoisted masks/flags
  const uint32_t ORIENT_INV = ~ORIENT_MASK;
  const uint32_t CLEAR_MASK = MASK_DISPLAY | ORIENT_MASK | VALUE_MASK_;
  const uint32_t FINAL_FLAGS = MASK_IS_SET;

  // Orientation 0..3: 0=left, 1=up, 2=right, 3=down
  const int delta[4] = {-size, -1, +size, +1};
```

```
21
22    // Pre-shifted orientation bits to avoid (o << SHIFT) in the loop
23    const uint32_t ORIENT_SHIFTED[4] = {
24      (0u << SHIFT), (1u << SHIFT), (2u << SHIFT), (3u << SHIFT)
25    };
26
27    uint64_t sum_path = 0;
28
29    for (int i = 0; i < count; ++i)
30    {
31      uint32_t* pos = arrayCell + baseIndex;
32      int path_length = 0;
33
34      for (;;)
35      {
36        // Single memory read
37        uint32_t cell = *pos;
38
39        // Stop if IsSet == 0
40        if ((cell & MASK_IS_SET) == 0u) break;
41
42        // Orientation 0..3, then wrap 0->1->2->3->0
43        uint32_t o = ((cell & ORIENT_MASK) >> SHIFT);
44        o = (o + 1u) & 3u;
45
46        // Write updated orientation (RMW on the read value)
47        *pos = (cell & ORIENT_INV) | ORIENT_SHIFTED[o];
48
49        // Advance
50        pos += delta[o];
51        ++path_length;
52      }
53
54      // First uninitialized cell -> merged final write
55      const uint32_t newValue = static_cast<uint32_t>(currentStart + i);
56
57      uint32_t cell = *pos;
58      cell &= ~CLEAR_MASK;        // bit31=0 + clear orient + value
59      cell |= FINAL_FLAGS;        // bit30=1
60      cell |= (newValue & VALUE_MASK_); // value
61      *pos = cell;
62
63      sum_path += static_cast<uint32_t>(path_length);
64    }
65
66    return static_cast<double>(sum_path) / static_cast<double>(count);
67  }
```

D Some definitions related to the Rotor-Router

Terms	Definitions
rotor-router model	Deterministic growth/walk model, a quasi-random analogue of random walks and IDLA.
Propp machine	Historical name of the rotor-router device, popularized by Jim Propp.
particle / chip	Mobile unit that moves by following the rotors.
rotor walk	Deterministic walk of a particle guided by the rotors.
rotor	Arrow (or local state) at a vertex pointing to a neighbor, which rotates in cyclic order.
cyclic order (of neighbors)	Predefined order in which the rotor rotates when activated.
(global) configuration	Positions of the particles and state of each rotor.
rotor configuration	Current assignment of directions/states to rotors.
source	Vertex where particles are injected.
deterministic random walk	Generic term for rotor walk: pseudo-random but deterministic trajectory.
quasi-random	Describes sequences or trajectories with small deviations from randomness.
discrepancy	Difference between the rotor-router distribution and that of an ideal random walk.
hitting time	Number of steps required to reach a set or a sink.
hitting measure	Distribution of absorption locations (compared with the harmonic measure).
rotor-router aggregation	Growth model where chips start from a source and stick to the first new site.
IDLA aggregation	Random analogue model, used as a theoretical comparison.
cluster	Set of sites ultimately occupied by the aggregation.
(cluster) boundary	Edge of the occupied region, subject of "limit shape" results.
limit shape	Asymptotic shape (often a disk in 2D) of the rescaled cluster.
Propp's disk	Conjecture/evidence that the 2D limit shape is a circle.
(boundary) fluctuations	Deviations of the boundary relative to the limit shape.

Z^d (integer lattice)	Standard regular grid used for experiments and theory.

Acknowledgments

The author is grateful to Pierre Gabriel (University of Tours, France) for his valuable comments and careful reading of the manuscript.

References

[1] Prof. Dr. Tobias Friedrich,The Rotor-Router Model

[2] Prof. James Propp,Site

[3] V. B. Priezzhev, Deepak Dhar, Abhishek Dhar, Supriya Krishnamurthy, Eulerian Walkers as a model of Self-Organised Criticality

[4] Jean-Paul Delahaye et Philippe Mathieu, Tous les chemins mènent au rond, 22 juillet 2015, POUR LA SCIENCE N° 454

[5] Lionel Levine and Yuval Peres, Strong spherical asymptotics for rotor-router aggregation and the divisible sandpile, Potential Analysis, vol. 30(1), pp. 1-27, 2009.

[6] T. Friedrich, Lionel Levine, Fast simulation of large-scale growth models, Potential Analysis, 4 June 2010.