

Efficient algorithm for the periodic Lorentz gas in 2 and 3 dimensions

Atahualpa S. Kraemer*

*Institut für Theoretische Physik II - Soft Matter Heinrich-Heine-Universität Düsseldorf
Building 25.32 Room O2.56 Universitätsstrasse 1 D-40225 Düsseldorf, Germany*

Nikolay Kryukov[†] and David P. Sanders[‡]

*Departamento de Física, Facultad de Ciencias, Universidad Nacional
Autónoma de México, Ciudad Universitaria, México D.F. 04510, Mexico*

(Dated: June 29, 2015)

We present an efficient algorithm to calculate collisions for a periodic Lorentz gas in two and three dimensions. The 2D algorithm uses continued fractions to obtain the exact disc with which a particle will collide at each step, instead of using periodic boundary conditions. The 3D version uses the 2D algorithm, but obtains these coordinates only with a probability p_{al} which depends on the size of the obstacles as $p_{al} \sim 1 - 0.5r^3$. We study the efficiency of the algorithm and apply it to calculate free path distributions. The algorithm can be generalized to higher dimensions.

I. INTRODUCTION

Lorentz gases are simple physical systems that present deterministic chaos [1], and are a popular model in statistical mechanics and nonlinear dynamics. This model consists of point particles that move freely until they encounter obstacles, often spheres, where they undergo elastic collisions.

These systems can have different configuration of the obstacles, e.g., random lattices [2–4] or quasiperiodic systems [5, 6]. However, because of its simplicity, the periodic case has been widely studied (see e.g. [7–10]); in this case, the model is equivalent to a Sinai billiard [7]. There are still many open questions in this area [11–14]. Many of the results obtained theoretically for these gases are in the limit where obstacles are very small, i.e., the so-called Boltzmann–Grad limit [15–23].

The standard simulation method for periodic Lorentz gases is to reduce to a single cell, with periodic boundary conditions, and, in the simplest case, an obstacle in the centre of the cell [24, 25]. However, this implies that the program must check each time that the particle crosses a border whether it will collide with the obstacle in the cell, or if it will move to the next cell. That implies solving at least two linear equations and one quadratic, then checking if the solution of the quadratic equation is real and finally taking the maximum value of the two or three results. If the obstacle is big enough, it is quite probable that the particle will collide each time it crosses the boundary. However, in the Boltzmann–Grad limit, this situation does not happen very often, and then much time is lost in unnecessary checks.

A suitable situation would be a formula to find the coordinates of the next obstacle giving the position and velocity of the particle. As we will see in section II this is indeed the first solution of the Diophantine inequality:

$$|\alpha q + b - p| \leq \frac{v^2}{v_x} r \quad (1)$$

where r is the radius of the obstacles, q and p are integer variables, and α and b are two real numbers related to the position \vec{x} and velocity \vec{v} of the particle. But this inequality is closely related to the best rational approximant to an irrational number for the case $b = 0$ and the solutions can be obtained by the continued fraction algorithm.

An algorithm for this, using continued fractions, was previously developed; see [26]. However, it was never published. Nevertheless, continued fractions have often been used to provide information about the free path distribution of the periodic Lorentz gas in the Boltzmann–Grad limit [8, 9, 15–21], and, relatively recently, Golse has developed an algorithm to code particles trajectories by using the continued fraction algorithm and the 3-length theorem [16].

However, this algorithm works a priori only if the particle leaves the surface of a disk. This restriction prevents the algorithm from being used in other geometries, such as quasiperiodic arrays, two incommensurate overlapping arrays of square lattices, or with different shapes of obstacles; such systems may produce a number of surprising effects [27]. On the other hand, due to the construction of the algorithm, using the 3-length theorem and continued fraction algorithm, it is not possible to use it in higher dimensions, which “is a notoriously more difficult problem” according to Golse.

II. 2D ALGORITHM

In this section, we present the 2D algorithm. Its goal is to calculate efficiently where the first collision of a particle will occur in a square lattice. Without loss of generality, we will use the lattice formed by the integer coordinates in the 2D plane. Then, we want to calculate the minimal time $t^* > 0$, such that a collision occurs with some disc with centre $\vec{c} = (q, p)$, with q and p integers. The condition is then $\|\vec{c} - (\vec{x}_0 + \vec{v}_0 t^*)\| = r$, where r is the radius of the obstacle, and \vec{x}_0 and

*ata.kraemer@gmail.com

[†]kryukov@ciencias.unam.mx

[‡]dpsanders@ciencias.unam.mx

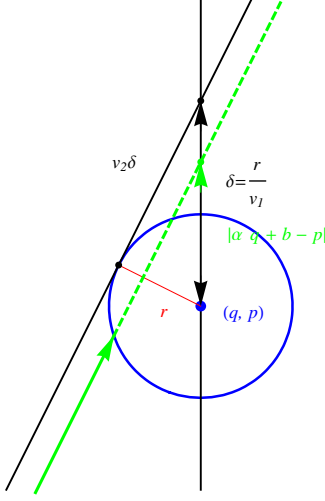


FIG. 1. Relation between the intersection of a line and a circle with integer coordinates and the intersection of the line $x = q$.

\vec{v}_0 are the initial position and velocity of the particle. In this case, the collision takes place at the point $\vec{x}_0 + \vec{v}_0 t^*$.

A. Collision with disc

The classical algorithm finds the intersection between a line corresponding to the trajectory of the particle and a circle. A first improvement follows from observing that instead we may look for the intersection with another line instead. We denote by (v_1, v_2) the components of the particle's velocity. Without loss of generality we take the speed to be 1, and $v_1 > 0$ and $v_2 > 0$ in the following; we can always rotate or reflect the system such that these conditions are satisfied, due to its symmetry.

We write the equation of the particle's trajectory as $y = \alpha x + b$ and look for its intersection with the vertical line $x = q$. As shown in figure 1, if $|\alpha q + b - p| = |b| < \delta := r/v_1$, then a collision will take place. Due to the periodic boundary conditions, we can always put $0 < b < 1$, so we only need to solve $b < \delta$. This gives us a tool to calculate with which obstacle the particle will collide.

Indeed, we do not need to apply periodic boundary conditions at every step, we only need to check

$$|\{\alpha q_n\} + b - 1| < \delta, \quad (2)$$

where $\{\cdot\}$ denotes the fractional part, and $q_n = q_{n-1} + 1$, where q_1 is the x -coordinate of the closest obstacle to the particle at $t = 0$. Then, the first q_n that satisfies this inequality will be q . To calculate p , we can note that always $p = \lfloor \alpha q + b \rfloor$ or $p = \lfloor \alpha q + b \rfloor + 1$

Now, to simplify even more the algorithm, consider the integer coordinates (q_n, p_n) such that

$$|\alpha q_n - p_n + b| < \delta, \quad (3)$$

and for any pair of numbers (i, j) such that $i < q_n$, then $|\alpha i - j + b| > \delta$, $q = q_n$, and $p = p_n$.

But $|\alpha q_i - p_i + b|$ are the distances between the integer coordinates (q_i, p_i) and the point $(q_i, \alpha q_i + b)$. Thus, we would like a sequence such that

$$|\alpha q_i - p_i + b| < |\alpha q_{i-1} - p_{i-1} + b| \quad (4)$$

for every $i > 1 \in \mathbb{N}$. Also, the first pair of integer coordinates q_0 and p_0 should be $(0, 0)$ or $(0, 1)$, minimizing $|\alpha q_0 - p_0 + b|$, that is:

$$|\alpha q_1 - p_1 + b| < f(b) = \begin{cases} b & \text{if } b < 1/2 \\ 1 - b & \text{if } b > 1/2 \end{cases}. \quad (5)$$

Note that $p_n = \lfloor \alpha q_n + b \rfloor = \begin{cases} \lfloor \alpha q_n \rfloor & \text{if } b + \alpha q_n - \lfloor \alpha q_n \rfloor < 1 \\ \lfloor \alpha q_n \rfloor + 1 & \text{if } b + \alpha q_n - \lfloor \alpha q_n \rfloor > 1 \end{cases}$ if $b < 1/2$ or

$$p_n = \lfloor \alpha q_n + b \rfloor + 1 = \begin{cases} \lfloor \alpha q_n \rfloor + 1 & \text{if } b + \alpha q_n - \lfloor \alpha q_n \rfloor < 1 \\ \lfloor \alpha q_n \rfloor + 2 & \text{if } b + \alpha q_n - \lfloor \alpha q_n \rfloor > 1 \end{cases}$$

if $b > 1/2$. Substituting the four cases from here in the two cases of equation 5, we obtain that indeed $p_1 = \lfloor \alpha q_1 \rfloor + 1$, and then, iterating inequality 4 we obtain

$$p_n = \lfloor \alpha q_n \rfloor + 1. \quad (6)$$

Mixing the inequality (3) and equation (6), we obtain again equation (2).

Thus, we have reduced the solution from two linear equations and one quadratic to one linear equation. Furthermore, now we do not check in every periodic cell, because if $\alpha > 1$, for every q_n we advance α cells. And we do not need to apply periodic boundary conditions until we reach the obstacle.

B. The Diophantine inequality: $|\alpha p - q| \leq \varepsilon$

Now, a better algorithm should find a way to find the set of q_i , such that inequality (4) holds for every i , and there is no integer q such that $q_i < q < q_{i-1}$ for some i and $|\{\alpha q_i\} + b - 1| < |\{\alpha q\} + b - 1| < |\{\alpha q_{i-1}\} + b - 1|$.

In order to do this, we can use the continued fraction algorithm to obtain solutions to the inequality $|\alpha q - p| \leq \varepsilon$. This algorithm already gives a sequence of (q_n, p_n) such that $|\alpha q_i - p_i| < |\alpha q_{i-1} - p_{i-1}|$ if $q_{i-1} < q_i$. So, if we turn our inequality (5) into this other inequality, we will find our algorithm just by using the continued fraction algorithm. Indeed, using equation 6 and the inequality 5, we obtain $|\{\alpha q_1\} - 1| < \begin{cases} 2b & \text{if } b < 1/2 \\ 2(1 - b) & \text{if } b > 1/2 \end{cases}$, which is almost the continued fraction inequality, except that p is always equal to $\lfloor \alpha q \rfloor + 1$.

$$|\alpha q - p| < \begin{cases} 2b & \text{if } b < 1/2 \\ 2(1 - b) & \text{if } b > 1/2 \end{cases} \quad (7)$$

Then, we can apply the continued fraction algorithm to obtain p and q of inequality (7). If $\lfloor \alpha q \rfloor + 1$, then, we have found

(q_1, p_1) , otherwise not, but we know that $p_1 \geq p$, and $q_1 \geq q$, so we can just use (q, p) even if they do not satisfy inequality 4, then calculate the sequence b_i as $b_0 = b$, $b_i = \{\alpha q_i + b\}$. If $b_n < \delta$ the algorithm stops, and the collision will take place with the obstacle centred at the coordinates (q_n, p_n) .

C. Explicit algorithm

Now we have all the necessary tools to implement the complete algorithm step by step. The Julia language [28] is used; the source code is available at ??

The function `frac(α, ϵ)` calculates the first integers (q, p) such that $|\alpha q - p| < \epsilon$ by using the continued fraction algorithm; see the appendix. This is used to calculate which disc is hit by the following function

Now, define the function `eff(m, b, r)` as follows:

```
function eff(m, b, r)
    kn = 0
    b1 = b
    epsilon = r*sqrt(m*m+1)
    if(b < epsilon || 1 - b < epsilon)
        if b < BigFloat("0.5")
            (q, p) = frac(m, 2.*b)
        else
            (q, p) = frac(m, 2.*(1. - b))
        end
    else
        b = mod(m*q + b, 1)
        kn += q
    end
    while b > epsilon && 1 - b > epsilon
        if b < BigFloat("0.5")
            (q, p) = frac(m, 2.*b)
        else
            (q, p) = frac(m, 2.*(1. - b))
        end
    end
    b = mod(m*q + b, 1)
    kn += q
    end
    q = kn
    p = int(m*q+b1)
    return [q, p]
end
```

This function applies the continued fraction algorithm along with the inequality 7 until the distance in the y-direction between the line and the integer coordinates (q, p) is less than ϵ .

Then, the algorithm to localize the first obstacle, with which a particle with velocity \vec{v} and position \vec{x} will collide, is as follows (in pseudocode):

```
if (v is in quadrant I) then
    (q,p)=efficient_algorithm(m, b, delta)
    p=int(m*q)+1
else if (v is in quadrant II) then
    m=-m
    % reflect the system with
```

```
%respect to y-axis
    (q,p)=efficient_algorithm(m, b, delta)
    p=int(m*q)+1
    m=-m
    % reflect the system with
    %respect to y-axis
    q=-q
    % reflect the system with
    %respect to y-axis
else if (v is in quadrant III) then
    b=1-b
    % reflect the system with
    %respect to the origin
    (q,p)=efficient_algorithm(m, b, delta)
    b=1-b
    % reflect the system with
    %respect to the origin
    p=-(int(m*q)+1)
    % reflect the system with
    %respect to the origin
    q=-q
    % reflect the system with
    %respect to the origin
else if (v is in quadrant IV) then
    b=1-b
    % reflect the system with
    %respect to the x-axis
    m=-m
    % reflect the system with
    %respect to the x-axis
    (q,p)=efficient_algorithm(m, b, delta)
    b=1-b
    % reflect the system with
    %respect to the x-axis
    m=-m
    % reflect the system with
    %respect to the x-axis
    p=-(int(m*q)+1)
    % reflect the system with
    %respect to the x-axis
endif
```

Using the technique mentioned above, which, for the initial position $(0, b)$, $0 < b < 1$ and for any initial velocity returns the integer coordinates of the obstacle of the first collision, we make the function `Lorentz(x, v, r)` that calculates the first collision of the particle in the 2D plane. In order to calculate the exact collision point and the velocity after the collision, we need to use the classical collision functions `collision` and `velo_col`.

III. 3D ALGORITHM

In this section, we present an efficient algorithm for calculating the next collision with a sphere in 3D on a simple cubic lattice. The algorithm works by projecting to 2D planes and using the 2D algorithm in each plane, as follows.

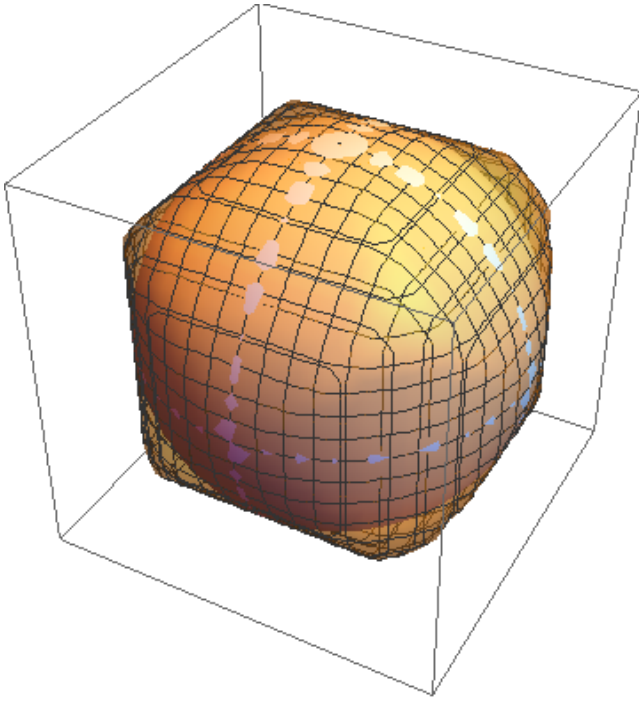


FIG. 2. A sphere of radius r embedded into the intersection of 3 orthogonal cylinders of the same radius. The volume where there is no sphere, but there is the intersection of the 3 cylinders, is the place where trajectories of particles can be considered as having collisions, when really they do not use the proposed algorithm in the 3D case.

Suppose we project a particle trajectory in a 3D lattice onto one of the x - y , x - z or y - z planes. We will obtain a periodic square lattice with a 2D trajectory, but with a speed that is, in general, not 1. Thus, we can apply the 2D algorithm for each of the 3 projections, and obtain the coordinates of 3 obstacles in the 3 projected planes where the first collision occurs.

We now check whether the obstacle coordinates in these projections correspond to the same 3D obstacle. If so, then we check if the particle really does collide with this obstacle by using the standard method; if it does, then we return the coordinates of the obstacle, and if not, we move the particle to the next cell and continue applying the algorithm. If the projections do not correspond to the same 3D obstacle, we move the particle to the cell containing the obstacle that is furthest away, and continue.

Using the 2D continued fraction algorithm in one of the planes xy , xz and yz is equivalent to calculating the first collision with a cylinder orthogonal to that plane. Joining those coordinates together is equivalent to calculating a collision with the intersection of the three orthogonal cylinders with the same radius.

Figure 2 shows the intersection of the 3 cylinders, together with a sphere of the same radius. The sphere is contained inside the intersection of the cylinders; the regions outside the sphere correspond to places where the 3D algorithm does not find the correct next collision, even when the three 2D obstacles coincide.

The probability that this occurs is the volume of the inter-

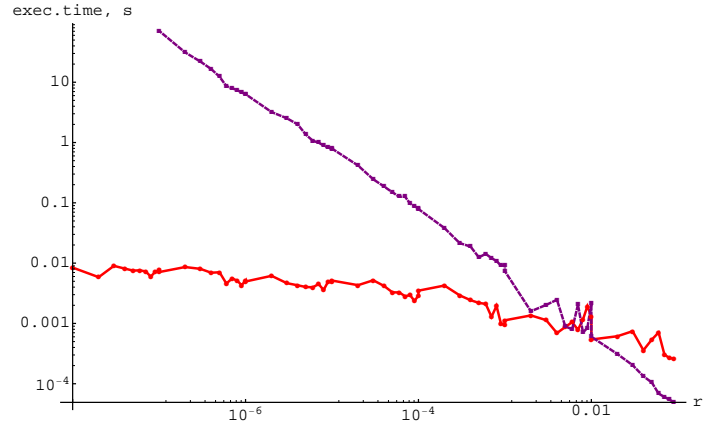


FIG. 3. Average execution time of finding the first collision in 2D Lorentz gas, for the classical (dotted curve) and efficient (solid curve) algorithms.

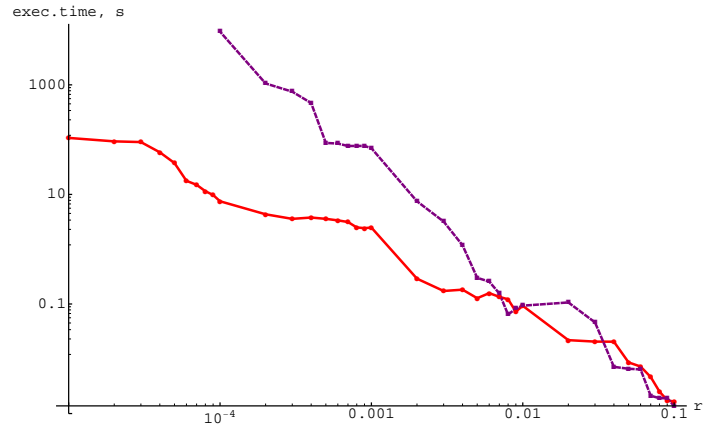


FIG. 4. Average execution time of finding the first collision in 3D Lorentz gas, for the classical (dotted curve) and efficient (solid curve) algorithms.

section of the cylinders minus the volume of the sphere:

$$p_{al} = 1 - (16 - \sqrt{128} - \frac{4}{3}\pi)r^3 \sim 1 - 0.5r^3. \quad (8)$$

This is very small for small obstacles; for example, obstacles of radius $r = 0.01$ give a probability $p_{al} \sim 0.999995$.

IV. NUMERICAL MEASUREMENTS

In order to test how efficient our algorithm is, we measured the average time of the execution of the function finding the first collision starting from the initial point around the origin, depending on the radius of the obstacle, for both the classical and efficient algorithms.

Figures 3 and 4 show the results for the 2D and 3D algorithms. As we can see, the new algorithm is significantly more efficient for $r < 0.01$.

Similarly, we calculated the execution time per cell depending on the radius of the obstacle, for both 2D and 3D efficient

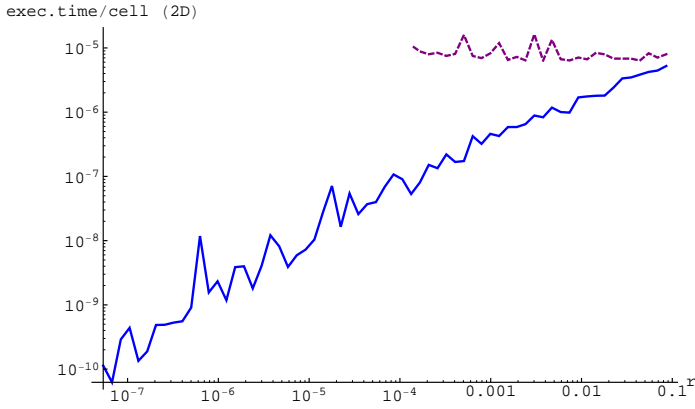


FIG. 5. Mean execution time per cell of finding the first collision in 2D Lorentz gas, for the classical (dashed curve) and efficient (solid curve) algorithms.

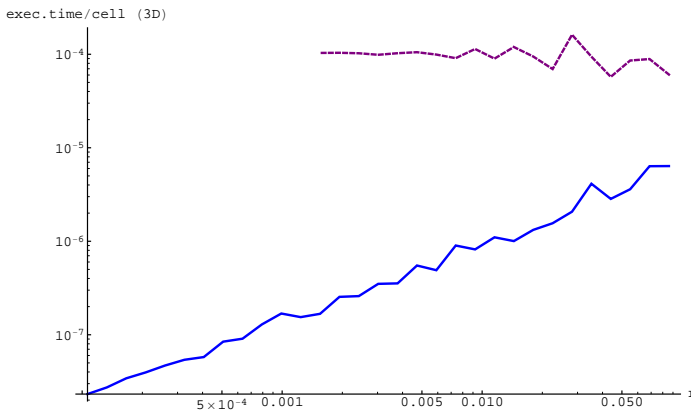


FIG. 6. Mean execution time per cell of finding the first collision in 3D Lorentz gas, for the classical (dashed curve) and efficient (solid curve) algorithms.

algorithm comparing it to classical. Classical algorithms use periodic boundary conditions, so the execution time per cell is about the same, regardless of the radius of the obstacle. For small radii, we can observe the efficiency of this algorithm.

A. Free flight distribution

As an example application of our algorithm, we have measured the distribution of free flight lengths for the first collision for certain systems studied by Marklof and Strömbergsson [27]. They proved the asymptotic decay of the free flight probability density incommensurable, overlapping periodic Lorentz gases in the Boltzmann–Grad limit.

Figure shows our numerical results for this distribution in the case of two and three overlapping lattices, compared to the rigorous result; the distributions obtained numerically do indeed follow the power laws predicted. The minimum time for each lattice is calculated separately, and the minimum of those results is then taken.

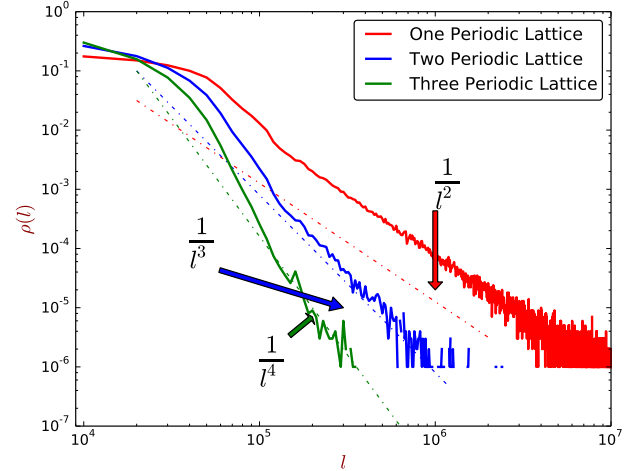


FIG. 7. Probability density of free flight length for 2 and 3 incommensurable, overlapping periodic Lorentz gases. The results for a single lattice are shown for comparison.

V. CONCLUSIONS

We have presented efficient algorithms to simulate periodic Lorentz gases in 2 and 3 dimensions when the obstacles are small and have compared the efficiency of these algorithms with the standard ones, showing that the relative efficiency indeed increases very fast in 2D and fast in 3D.

The 3D algorithm can readily be generalized to higher dimensions. This work is in progress. We expect that these algorithms will be useful to continue the study of these models.

VI. ACKNOWLEDGMENTS

ASK received support within the Emmy Noether program (Grant Schm 2657/2). NK thanks DGAPA-UNAM for a post-doctoral fellowship. DPS received financial support from CONACYT grant CB-101246 and DGAPA-UNAM PAPIIT grants IN116212 and IN117214.

VII. APPENDIX: APPROXIMATION OF AN IRRATIONAL NUMBER BY A RATIONAL

In this section we will summarize some of the results about continued fractions used in the algorithm. The proofs can be found in books on number theory, e.g., [29]). The geometrical interpretation was also suggested before by many other authors; see, for example, [30].

We define a continued fraction as follows: A continued fraction is an expression obtained through an iterative process of representing a number α as the sum of its integer part a_0 and the reciprocal of another number $\alpha_1 = \alpha - a_0$, then writing α_1 as the sum of its integer part a_1 and the reciprocal of

$\alpha_2 = \alpha_1 - a_1$, and so on:

$$\alpha = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}}$$

This expression produces a sequence of integers $[\alpha] = a_0$, $[\alpha_1] = a_1$, $[\alpha_2] = a_2$, etc. Then, we define inductively two sequences of integers $\{p_n\}$ and $\{q_n\}$ as follows:

$$p_{-2} = 0, \quad p_{-1} = 1, \quad p_i = a_i p_{i-1} + p_{i-2}; \quad (9)$$

$$q_{-2} = 1, \quad q_{-1} = 0, \quad q_i = a_i q_{i-1} + q_{i-2}. \quad (10)$$

With this sequence we can approximate any irrational number α using the Hurwitz theorem: For every irrational number α all the relative prime integers p_n, q_n of the sequences defined in equations 9 and 10 satisfy

$$|\alpha - \frac{p_n}{q_n}| \leq \frac{1}{q_n^2}. \quad (11)$$

-
- [1] P. Cvitanović, P. Gaspard, and T. Schreiber, *Chaos: An Interdisciplinary Journal of Nonlinear Science* **2**, 85 (1992).
 - [2] A. Latz, H. Van Beijeren, and J. Dorfman, *Physical review letters* **78**, 207 (1997).
 - [3] C. Dellago and H. Posch, *Physical review letters* **78**, 211 (1997).
 - [4] H. Van Beijeren, A. Latz, and J. Dorfman, *Physical Review E* **57**, 4077 (1998).
 - [5] A. S. Kraemer and D. P. Sanders, *Physical review letters* **111**, 125501 (2013).
 - [6] B. Wennberg, *Journal of Statistical Physics* **147**, 981 (2012).
 - [7] L. A. Bunimovich and Y. G. Sinai, *Communications in Mathematical Physics* **78**, 479 (1981).
 - [8] P. Bleher, *Journal of statistical physics* **66**, 315 (1992).
 - [9] N. Chernov, *Journal of Statistical Physics* **74**, 11 (1994).
 - [10] T. Gilbert, H. C. Nguyen, and D. P. Sanders, *Journal of Physics A: Mathematical and Theoretical* **44**, 065001 (2011).
 - [11] T. Gilbert and D. P. Sanders, *Physical Review E* **80**, 041121 (2009).
 - [12] J. Marklof and A. Strömbergsson, *Geometric and Functional Analysis* **21**, 560 (2011).
 - [13] P. Nandori, D. Szasz, and T. Varju, *arXiv preprint arXiv:1210.2231* (2012).
 - [14] C. P. Dettmann, *Journal of Statistical Physics* **146**, 181 (2012).
 - [15] E. Caglioti and F. Golse, *Communications in mathematical physics* **236**, 199 (2003).
 - [16] F. Golse, in *Nonlinear Partial Differential Equations* (Springer, 2012), pp. 39–99.
 - [17] F. P. Boca and A. Zaharescu, *Communications in mathematical physics* **269**, 425 (2007).
 - [18] F. Golse, in *Proceedings of the International Congress of Mathematicians (Madrid, 2006)* (2006), vol. 3, pp. 183–20.
 - [19] E. Caglioti and F. Golse, *Comptes Rendus Mathematique* **346**, 477 (2008).
 - [20] E. Caglioti and F. Golse, *Journal of Statistical Physics* **141**, 264 (2010).
 - [21] F. Golse and B. Wennberg, *ESAIM: Mathematical Modelling and Numerical Analysis* **34**, 1151 (2000).
 - [22] J. Marklof and A. Strömbergsson, *Nonlinearity* **21**, 1413 (2008).
 - [23] J. Bourgain, F. Golse, and B. Wennberg, *Communications in mathematical physics* **190**, 491 (1998).
 - [24] D. P. Sanders, *Physical Review E* **71**, 016220 (2005).
 - [25] D. P. Sanders, *Physical Review E* **78**, 060101 (2008).
 - [26] A. Zacherl, T. Geisel, J. Nierwetberg, and G. Radons, *Physics Letters A* **114**, 317 (1986).
 - [27] J. Marklof and A. Strömbergsson, *Journal of Statistical Physics* **155**, 1072 (2014).
 - [28] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *CoRR abs/1411.1607* (2014), URL <http://arxiv.org/abs/1411.1607>.
 - [29] I. Niven, H. S. Zuckerman, and H. L. Montgomery, *An introduction to the theory of numbers* (John Wiley & Sons, 2008).
 - [30] A. Nogueira, *Israel Journal of Mathematics* **90**, 373 (1995).