

Full Euclidean Algorithm by Means of a Steady Walk

Carlos M. Falcon Rodriguez¹, Maria A. Garcia Cruz¹, Claudia Falcon²

¹Grupo de Investigación de Análisis Matemático, Instituto de Formación docente Salomé Ureña, Santo Domingo, D.N., República Dominicana

²Wake Forest University, Winston-Salem, NC, USA

Email: falc4847@gmail.com, mcelmdf@gmail.com, falconc@wfu.edu

How to cite this paper: Falcon Rodriguez, C.M., Cruz, M.A.G. and Falcon, C. (2021) Full Euclidean Algorithm by Means of a Steady Walk. *Applied Mathematics*, 12, 269-279.

<https://doi.org/10.4236/am.2021.124018>

Received: March 3, 2021

Accepted: April 11, 2021

Published: April 14, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Let x and y be two positive real numbers with $x < y$. Consider a traveler, on the interval $[0, y/2]$, departing from 0 and taking steps of length equal to x . Every time a step reaches an endpoint of the interval, the traveler rebounds off the endpoint in order to complete the step length. We show that the footprints of the traveler are the output of a full Euclidean algorithm for x and y , whenever y/x is a rational number. In the case that y/x is irrational, the algorithm is, theoretically, not finite; however, it is a new tool for the study of its irrationality.

Keywords

Extended Euclidean Algorithm, Greatest Common Divisor, Incommensurable Numbers, Steady Walk, Diophantine Equation

1. Introduction

The Euclidean greatest common divisor (gcd) algorithm is one of the most successful algorithms in Mathematics. Its improvements, like Lehmer's gcd algorithm [1], are currently being used in cryptography protocols to factorize large composite numbers [2]. Examples of gcd 's application in Number Theory are: the Diophantine equations [3], the Chinese remainder theorem [4], and continued fractions [5]. The gcd algorithm also inspired Gabriel Lamé in the creation of the computational complexity theory [6]. Euclid gave expression to the gcd algorithm in the propositions of Book Seven of Euclid's Elements [7]. He used concepts about divisibility and relatively prime numbers that may have taken the focus away from the case when the numbers are not integers. However, it is known that the algorithm works with real numbers, even though

there are not many current references on this topic. But in fact, the concept of *gcd* has historically also used the name “greatest common measure” [7], indicating the measurement of real magnitudes.

In this paper, we present an algorithm that finds inspiration in Euclid’s *gcd* algorithm and Eudoxus’ proportion theory [8]. The impeccable theory of proportions of Eudoxus dates some decades before Euclid and uses the simple idea of fitting integer multiples of two real numbers with the intention of defining their ratio. In the next paragraphs, we provide some preliminary definitions and review useful background for later reference.

Let x and y be positive real numbers. Considering the fraction y/x , we distinguish two cases, the case where y/x is a rational, and the case where y/x is an irrational. If y/x is a rational, we denote $SF(y/x)$ as the unique simplified fraction a/b , with a and b natural numbers and $gcd(a,b)=1$, such that

$$\frac{y}{x} = SF(y/x). \quad (1)$$

The positive real number z is a divisor of x whenever there exists a natural number k fulfilling $kz = x$. Each time the set of common divisors of x and y is not empty, the greatest common divisor of x and y , $gcd(x,y)$ is the maximum z that is a divisor of x and is also a divisor of y . In the case that the set of common divisors of x and y is empty, we say that x and y are incommensurable.

The relation between the previous definitions of SF and gcd , is that when z is the greatest common divisor of x and y , and $kz = x$, $mz = y$, we have that $mkz = mx$, $kmz = ky$. Consequently,

$$\left(\frac{m}{k}x = y\right) \wedge \left(\frac{k}{m}y = x\right), \quad (2)$$

and

$$\left(\frac{m}{k} = SF\left(\frac{y}{x}\right)\right) \wedge \left(\frac{k}{m} = SF\left(\frac{x}{y}\right)\right). \quad (3)$$

The veracity of (3) is a consequence of the simple fact that if $gcd(k,m) = d$ and $d > 1$, then the $gcd(x,y)$ would be equal to zd instead of z .

The previous definition of $gcd(x,y)$ for x and y reals is not new. Indeed, the original version of the Euclidean algorithm works with quantities represented by segments. Its inspiration was to measure one segment with another. The algorithm that we present in Section 2 recovers this original meaning, which can be insightful even for children in elementary school.

The extended Euclidean algorithm, described in [9], has many applications in the solution of Diophantine equations [10] [11]. Improving its performance is a frequent subject in publications [12]. In all the rediscoveries and improvements that we can find since the 5th-6th century to our days, the Euclid’s way to find *gcd* is the cornerstone. The algorithm we present in Section 2 is not only a way

to get the solution of the Diophantine equation,

$$mx + ny = \gcd(x, y), \quad (4)$$

but it also provides a complete arithmetic in $x\mathbb{Z}/y\mathbb{Z}$, the ring of the residue classes modulo y . The “steady walk” algorithm (*swa*) described in this paper does not use the Euclidean algorithm for \gcd . The proof we give in Section 2 is interesting by itself and uses the following not common formula of $\tan(n\theta)$ as a function of $\tan(\theta)$, when $\tan(\theta) \neq \infty$ and $n \geq 1$,

$$\tan(n\theta) = \frac{\sum_{k=0}^{\lfloor (n-1)/2 \rfloor} (-1)^k \binom{n}{2k+1} \tan^{2k+1}(\theta)}{\sum_{k=0}^{\lfloor n/2 \rfloor} (-1)^k \binom{n}{2k} \tan^{2k}(\theta)}. \quad (5)$$

The formula (5) can be proven by mathematical induction, using the tangent of the sum of two angles. The integers $\lfloor (n-1)/2 \rfloor$ and $\lfloor n/2 \rfloor$ are the floor function values on $(n-1)/2$ and $n/2$ respectively. Let

$$P_n(x) = \sum_{k=0}^{\lfloor (n-1)/2 \rfloor} (-1)^k \binom{n}{2k+1} x^{2k+1}. \quad (6)$$

Consequently, in the case that $\tan(x) \neq \infty$, the formula (5) implies that

$$\tan(nx) = 0 \Leftrightarrow P_n(\tan(x)) = 0. \quad (7)$$

The rest of this article is organized in the following way. Section 2 explains the *swa* and presents its formal description with the necessary proof. In Section 3, the implementation of the algorithm provides interesting resulting examples. And finally, we present a concluding section.

2. The Algorithm

Let x and y be positive real numbers, with $x < y$. The *swa* consists in marking, between two “walls”, consecutive steps, with step length equal to x . The walls are at 0 and at $y/2$ of the real line. The first step begins at the 0 wall. If a step has not finished when it meets a wall, it rebounds off the wall in order to complete the step length. The walk only changes direction when the steps meet and rebound off the walls.

Formalizing the previous description, we write,

$$f : [0, +\infty[\setminus \{(ky/2) : k \in \mathbb{Z}, k > 0\} \rightarrow [0, y/2],$$

defined by: $f(0) = 0$ and

$$f(t) = \begin{cases} t - ny/2, & t \in \left(\frac{ny}{2}, \frac{(n+1)y}{2}\right) \wedge n \text{ is even} \\ (n+1)y/2 - t, & t \in \left(\frac{ny}{2}, \frac{(n+1)y}{2}\right) \wedge n \text{ is odd.} \end{cases} \quad (8)$$

Then the mark of the m -step is $f(mx)$.

Theorem 1. If the set

$$F = \{s \in \mathbb{N} : (\exists t \in \mathbb{N}, t < s, f(sx) = f(tx)) \vee (f(sx) = y/2)\} \quad (9)$$

is not empty and we call $k + 1 = \min(F)$, then the following statements are true:

- 1) The set $\{f(mx) : m \in [0, k]\}$ is an array of equidistant points.
- 2) The distance between contiguous points is the $\gcd(x, y) = d$.
- 3) The distance between the maximum of $\{f(mx) : m \in [0, k]\}$ and $y/2$ is equal to d or $d/2$.
- 4) If b is the number of points that are less than the first step, then $bd = x$. Additionally, $y = (2k + 2)d \vee y = (2k + 1)d$, in correspondence to the disjunction in item 3, respectively.
- 5) $(2k + 2)/b = SF(y/x) \vee (2k + 1)/b = SF(y/x)$, in correspondence to the above disjunction respectively.

6) If the point id of the array, corresponds to the m_i -step, then,

$$m_i b \equiv \pm i \pmod{(2k + 2)} \vee m_i b \equiv \pm i \pmod{(2k + 1)},$$

in correspondence to the above disjunction respectively.

Proof. Let us visualize the algorithm in the following way:

Construct a circle of radius equal to y/π . In this way the semicircle has a length equal to y , and the quarter circle is $y/2$ long. The steps will go from the point of zero angle, counterclockwise and each step has length x on the circle. If mx is the m -step, the angle in radians of this point is $m x \pi / y$. We denote the before angle as $\angle x_m$. There is only one angle, $\angle \tilde{x}_m$ in the first quadrant fulfilling,

$$\tan^2(\angle x_m) = \tan^2(\angle \tilde{x}_m). \quad (10)$$

Simple observation shows that the steps \tilde{x}_m map on the steps $f(mx)$ under the natural bijection between the segment $[0, y/2]$ and the first quarter circle. In the case that there exists $t, t < s$, with $f(sx) = f(tx)$, we have that $\tilde{x}_s = \tilde{x}_t$ and consequently,

$$(sx\pi/y = \pm tx\pi/y + m\pi, m \in \mathbb{Z}) \quad (11)$$

\Updownarrow

$$(sx = \pm tx + my, m \in \mathbb{Z}) \quad (12)$$

\Updownarrow

$$\left(\frac{s \mp t}{m} x = y, m \in \mathbb{Z}\right). \quad (13)$$

The last equality (13) is a consequence of $(s \mp t \neq 0 \Rightarrow m \neq 0)$. Hence, we can suppose that there exist a and b , fulfilling

$$\frac{a}{b} x = y, \text{ with } a, b \in \mathbb{N} \text{ and } \gcd(a, b) = 1. \quad (14)$$

The case $f(sx) = y/2$ is equivalent to

$$sx = 2ky + y/2, \quad k \in \mathbb{N}. \tag{15}$$

Similarly, we conclude that there exist a and b , fulfilling,

$$\frac{a}{b}x = y, \quad \text{with } a, b \in \mathbb{N} \text{ and } \gcd(a, b) = 1. \tag{16}$$

From (16), we obtain

$$\tan\left(a\angle\frac{nx}{b}\right) = \tan\left(\angle a\frac{nx}{b}\right) = \tan(n\pi) = 0, \quad n \in \mathbb{N}. \tag{17}$$

Let $A = \{1, 2, \dots, \lfloor (a-1)/2 \rfloor\}$. It is easy to verify that,

- 1) $n \in A \Rightarrow (\tan(\angle(nx/b)) \neq 0 \wedge \tan(\angle(nx/b)) \neq \infty)$,
- 2) $(n, m \in A \wedge n \neq m) \Rightarrow \tan^2(\angle(nx/b)) \neq \tan^2(\angle(mx/b))$,

and,

$$P_a(\tan(\angle(nx/b))) = 0, \quad n \in A. \tag{18}$$

The set of equalities (18) can be written as

$$\sum_{k=0}^{\lfloor (a-1)/2 \rfloor} (-1)^k \binom{a}{2k+1} (\tan(\angle(nx/b)))^{2k+1} = 0, \quad n \in A. \tag{19}$$

Dividing each equation by $\tan(\angle(nx/b)) \neq 0$, we get

$$\sum_{k=0}^{\lfloor (a-1)/2 \rfloor} (-1)^k \binom{a}{2k+1} (\tan^2(\angle(nx/b)))^k = 0. \tag{20}$$

Denoting $x_k = (-1)^k \binom{a}{2k+1}$, we have,

$$\sum_{k=0}^{\lfloor (a-1)/2 \rfloor} (\tan^2(\angle(nx/b)))^k x_k = 0, \quad n \in A. \tag{21}$$

We can interpret the set of equalities (21) as a system of linear equations with unknown variables $\{x_k\}$. This is a system where the number of variables exceeds by one the number of equations.

We now observe that, if for some $m \in A$ we have

$$\tan(\angle(mx)) \neq 0 \wedge \tan(\angle(mx)) \neq \infty \tag{22}$$

and

$$\tan^2(\angle(mx)) \neq \tan^2(\angle(nx/b)), \quad \text{for all } n \in A, \tag{23}$$

we also have

$$P_a(\tan(\angle mx)) = \tan(a\angle(mx)) = \tan(\angle(amx)) = \tan(bm\pi) = 0. \tag{24}$$

Then, we would have the square system,

$$\sum_{k=0}^{\lfloor (a-1)/2 \rfloor} (\tan^2(\angle(nx/b)))^k x_k = 0, \quad n \in A, \tag{25}$$

$$\sum_{k=0}^{\lfloor (a-1)/2 \rfloor} (\tan^2(\angle(mx)))^k x_k = 0, \tag{26}$$

whose determinant is equal to a non-null Vandermonde determinant. This is a contradiction due to the existence of the non-null solution $\{x_k\}$.

Let us examine what is failing. Checking (22),

$$\begin{aligned} (\tan(\angle(mx)) = 0 \vee \tan(\angle(mx)) = \infty) &\Rightarrow mx = ky/2, \quad k \in \mathbb{Z} & (27) \\ &\Rightarrow 2mb = ka, \quad k \in \mathbb{Z}. \end{aligned}$$

The last equality in (27) implies that a divides $2m$. This is impossible because $m \in A$. Hence, the only possibility is that (23) fails. In other words, for each $m \in A$, there exist $n \in A$ fulfilling,

$$\tan^2(\angle(mx)) = \tan^2(\angle(nx/b)). \tag{28}$$

Also, it is easy to check that the values $\{\tan^2(\angle(mx)) / m \in A\}$ are different,

$$\begin{aligned} \tan^2(\angle(mx)) = \tan^2(\angle(m_1x)) &\Rightarrow mx = \pm m_1x + ky & (29) \\ &\Rightarrow b(m \mp m_1) = ka. \end{aligned}$$

From (29), we have that $m \mp m_1 = 0$, which clearly implies $m = m_1$. Therefore, by the Pigeonhole Principle, there exists a bijective function f such that

$$\begin{aligned} f : A &\rightarrow A & (30) \\ n &\rightarrow m_n, \end{aligned}$$

is defined by the property,

$$\tan^2(\angle(m_nx)) = \tan^2(\angle(nx/b)). \tag{31}$$

Now, we observe that,

$$k = \text{Cardinality}(A) = \left\lfloor \frac{a-1}{2} \right\rfloor = \begin{cases} (a-2)/2 & \text{with } a \text{ even} \\ (a-1)/2 & \text{with } a \text{ odd,} \end{cases} \tag{32}$$

and $d = x/b$, so the first five points of the theorem are straightforward. Recapitulating,

1) The set $\{f(mx) : m \in [0, k]\}$ corresponds to the set of steps in the first quadrant $\{\tilde{x}_m\}$, and this last set corresponds to the angles $\{\angle(nx/b) : n \in A\}$.

2) $d = x/b$ is the arc length between contiguous points of the array. Furthermore,

$$a\left(\frac{x}{b}\right) = y \wedge b\left(\frac{x}{b}\right) = x. \tag{33}$$

3) When a is even, the maximum of the array of steps is

$$\left(\frac{a-2}{2}\right)\left(\frac{x}{b}\right) = \left(\frac{1}{2}\right)\left(\frac{a}{b}\right)x - d = \left(\frac{y}{2}\right) - d. \tag{34}$$

And when a is odd, the maximum is

$$\left(\frac{a-1}{2}\right)\left(\frac{x}{b}\right) = \left(\frac{1}{2}\right)\left(\frac{a}{b}\right)x - \frac{d}{2} = \left(\frac{y}{2}\right) - \frac{d}{2}. \tag{35}$$

4) The equality $bd = x$ is a consequence of item 2. Additionally, when a is even,

$$\left(\frac{a-2}{2}\right) = k, \quad (36)$$

and $y = ad = (2k+2)d$. When a is odd,

$$\left(\frac{a-1}{2}\right) = k, \quad (37)$$

and $y = ad = (2k+1)d$.

5) In the case a is even,

$$\frac{a}{b} = \frac{2k+2}{b} = SF(y/x). \quad (38)$$

And when a is odd,

$$\frac{a}{b} = \frac{2k+1}{b} = SF(y/x). \quad (39)$$

Finally, let us prove item 6,

$$\tan^2(\angle(m_i, x)) = \tan^2(\angle(ix/b)) \quad (40)$$

↓

$$m_i x = \pm(ix/b) + my, \quad m \in \mathbb{Z} \quad (41)$$

↓

$$m_i \frac{by}{a} \mp i \frac{y}{a} = my, \quad m \in \mathbb{Z} \quad (42)$$

↓

$$m_i b \mp i = ma, \quad m \in \mathbb{Z}. \quad (43)$$

The previous sequence of implications (40)-(43) finishes the proof of the theorem. \square

Note that the equality (43) is equivalent to

$$m_i x \mp i \frac{x}{b} = my, \quad m \in \mathbb{Z}, \quad (44)$$

and this last equality (44) justifies that the swa provides a complete arithmetic in $x\mathbb{Z}/y\mathbb{Z}$, as stated in the introduction.

3. Implementation and Examples

In this section, we show the process of the algorithm described in Section 2 as a pseudocode. The examples and figures were implemented using the python programming language. In the figures, we mark the steps corresponding to each complete path of $[0, y/2]$ on different segments of a polygonal curve. See **Figures 1-3**. The geometrical behavior displayed in the plots change with the two numbers, x and y , in the **Algorithm 1** and with the number of iterations N .

Algorithm 1. Steady walk algorithm.

```

procedure STEPS( $x, y, N$ )
 $\triangleright x$  and  $y$  positive real numbers  $x < y$ 
 $\triangleright N$  is a natural number, max number of iterations
  Output : vector  $\vec{P}$ , of  $n \leq N$  steps.
  initialize parameters,  $S = 1, i = 1, P[0] = 0$ 
  while  $i \leq N$  do
    if  $S = 1$  then step forward,  $p = P[i - 1] + x$ 
      if wall is reached ( $p = y/2$  or  $p = y$ ) then break
      else if  $p > y$  then  $p = p - y$ 
      else if wall is surpassed ( $y/2 < p < y$ ) then
        walk backwards ( $S = -1, p = y - p$ )
      end if

    else if  $S = -1$  then step backwards,  $p = P[i - 1] - x$ 
      if wall is reached ( $p = 0$  or  $p = -y/2$ ) then break
      else if  $0 < p < y/2$  then  $p = p + y$ 
      else if wall is surpassed ( $-y/2 < p < 0$ ) then
        walk forward ( $S = 1, p = -p$ )
      end if
    end if

    if  $p$  is already in vector  $\vec{P}$  then break
    end if
     $P[i] = p$ , increment  $i, i = i + 1$ 
  end while

  return  $\vec{P}$ 
end procedure

```

Example 1: The steady walk algorithm (*swa*) for two integers, $x = 143$ and $y = 847$, with $N = 200$ iterations. As seen in **Figure 1**, the algorithm stops at the iteration $(847/2)/11$. In this case the Euclidean algorithm for *gcd* only needs two divisions and the *swa* obtains two footprints, 286 and 275, with difference equal to the *gcd* after two sums and two differences.

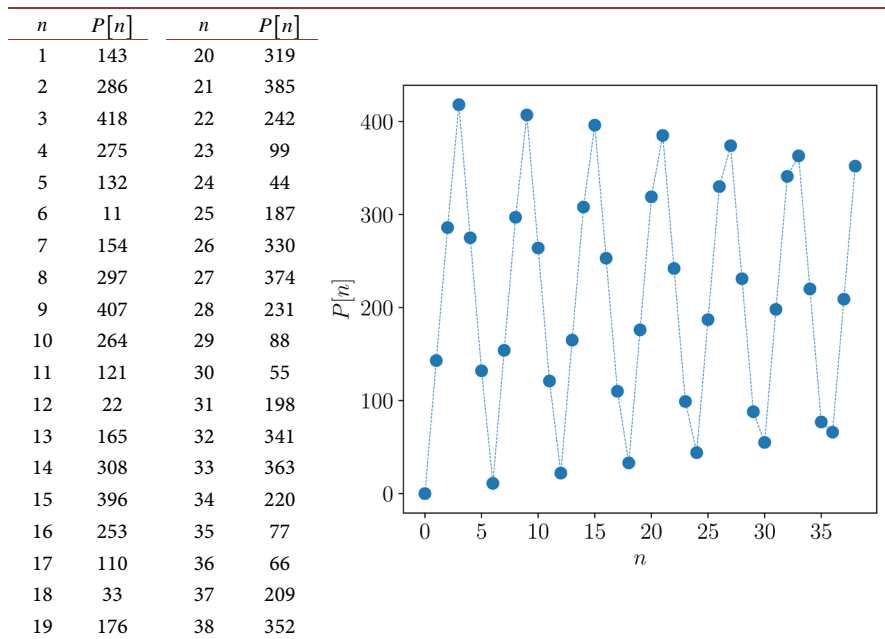


Figure 1. Output of *swa* for $x = 143$ and $y = 847$. The plot shows the steps $P[n]$ vs. the iteration number n . The table lists the numerical values of all points.

Example 2: The *swa* for the numbers $x = 1$ and the irrational $y = \pi$, with $N = 55$ iterations. These two numbers x and y are incommensurable. We choose this particular number of iterations N so that it shows interesting regular patterns (see **Figure 2**).

n	$P[n]$	n	$P[n]$	n	$P[n]$	n	$P[n]$
0	0	14	1.4336	28	0.27433	42	1.1593
1	1	15	0.70796	29	0.72567	43	0.9823
2	1.1416	16	0.29204	30	1.4159	44	0.0177
3	0.14159	17	1.292	31	0.41593	45	1.0177
4	0.85841	18	0.84956	32	0.58407	46	1.1239
5	1.2832	19	0.15044	33	1.5575	47	0.12389
6	0.28319	20	1.1504	34	0.55752	48	0.87611
7	0.71681	21	0.99115	35	0.44248	49	1.2655
8	1.4248	22	0.00885	36	1.4425	50	0.26548
9	0.42478	23	1.0089	37	0.69911	51	0.73452
10	0.57522	24	1.1327	38	0.30089	52	1.4071
11	1.5664	25	0.13274	39	1.3009	53	0.40708
12	0.56637	26	0.86726	40	0.8407	54	0.59292
13	0.43363	27	1.2743	41	0.1593	55	1.5487

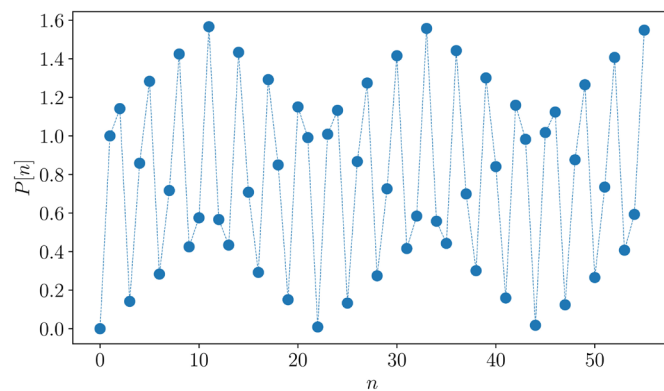


Figure 2. Output of *swa* for $x = 1$ and $y = \pi$. The plot shows the steps $P[n]$ vs. the iteration number n . The table lists the numerical values of all points.

Example 3: **Figure 3** shows the *swa* for two irrational numbers, $x = \sqrt{2}$ and $y = e$, with $N = 200$ iterations.

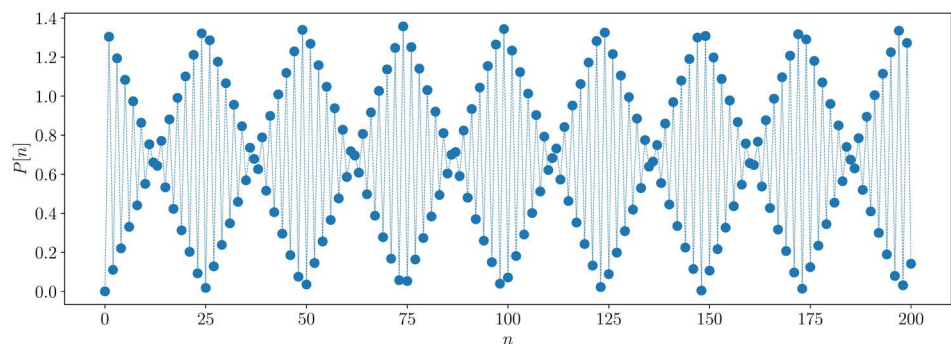


Figure 3. Output of *swa* for $x = \sqrt{2}$ and $y = e$. The plot shows the steps $P[n]$ vs. the iteration number n .

4. Conclusion

The full Euclidean algorithm that we present in this paper has advantages in several aspects. As a result of its graphical output, it can be a very useful tool in the mathematical explanations about incommensurability, irrationality, great common divisor, solution of the Diophantine Equation (4) and arithmetic in the ring \mathbb{Z}_a , of residue classes modulo a . Another advantage is the improvement of the extended Euclidean algorithm. The process can be expedited since the difference between any two footprints is a new candidate to redirect the algorithm toward a shorter path. In fact, to begin with $y/2$, instead of y , is already a clear improvement. The last aspect we would like to mention is the study of irrationality. The distribution of footprints in $[0, y/2]$, when the number of steps increases, is a promising tool in the understanding of different kinds and grades of irrationality, and the Diophantine approximation. Some related results can be found in [13]. Finally, *swa* is conceptually easier than the extended and classic Euclidean algorithms, allowing many shortcuts that improve its efficiency.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Lehmer, D.H. (1938) Euclid's Algorithm for Large Numbers. *The American Mathematical Monthly*, **45**, 227-233. <https://doi.org/10.1080/00029890.1938.11990797>
- [2] Durand, A. (1998) Efficient Ways to Implement Elliptic Curve Exponentiation on a Smart Card. *International Conference on Smart Card Research and Advanced Applications*, Springer, Berlin, 357-365. https://doi.org/10.1007/10721064_33
- [3] Mordell, L.J. (1969) *Diophantine Equations*. Academic Press, London, New York.
- [4] Childs, L.N. (2009) A Concrete Introduction to Higher Algebra. ch. The Chinese Remainder Theorem. Springer-Verlag, New York, 253-281. <https://doi.org/10.1007/978-0-387-74725-5>
- [5] Khinchin, A.Y. (1997) *Continued Fractions*, Russian ed. Dover Publications, New York.
- [6] Lamé, G. (1844) Note sur la limite du nombre des divisions dans la recherche du plus grand commun diviseur entre deux nombres entiers. *Comptes Rendus Acad. Sci.*, Paris.
- [7] Heiberg, J.L. (2007) *Euclid's Elements*. Richard Fitzpatrick, TX, USA.
- [8] Nikolic, M. (1974) The Relation between Eudoxus' Theory of Proportions and Dedekind's Theory of Cuts. In: Cohen, R.S., Stachel, J.J. and Wartofsky, M.W., Eds., *For Dirk Struik*, Springer, Dordrecht, 225-243. https://doi.org/10.1007/978-94-010-2115-9_19
- [9] Knuth, D.E. (2014) *Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley Professional, Boston.
- [10] Ramachandran, P. (2006) Use of Extended Euclidean Algorithm in Solving a System of Linear Diophantine Equations with Bounded Variables. In: Hess, F., Pauli, S. and Pohst, M., Eds., *Algorithmic Number Theory. ANTS 2006. Lecture Notes in Computer Science*, Vol 4076. Springer, Berlin, Heidelberg, 182-192.

https://doi.org/10.1007/11792086_14

- [11] Rankin, S. (2013) The Euclidean Algorithm and the Linear Diophantine Equation $ax + by = \gcd(a, b)$. *The American Mathematical Monthly*, **120**, 562-564.
<https://doi.org/10.4169/amer.math.monthly.120.06.562>
- [12] Iliev, A. and Kyurkchiev, N. (2018) The Faster Euclidean Algorithm. *Collection of Scientific Works from Conference*, Pamporovo, Bulgaria, 28-30.
- [13] Niven, I. (1956) Irrational Numbers. (Carus Mathematical Monographs, Series Number 11). John Wiley and Sons, Inc., New York.
<https://doi.org/10.5948/9781614440116>