





Article

# Some New Symbolic Algorithms for the Computation of Generalized Asymptotes

Elena Campo-Montalvo <sup>1,\*</sup>, Marián Fernández de Sevilla <sup>2,†</sup>, J. Rafael Magdalena Benedicto <sup>3,†</sup>  
and Sonia Pérez-Díaz <sup>4,†</sup>

<sup>1</sup> Department of Computer Engineering, University of Alcalá, 28871 Alcalá de Henares, Spain

<sup>2</sup> Department of Computer Science, University of Alcalá, 28871 Alcalá de Henares, Spain

<sup>3</sup> Department of Electronic Engineering, University of Valencia, 46010 València, Spain

<sup>4</sup> Department of Physics and Mathematics, University of Alcalá, 28871 Alcalá de Henares, Spain

\* Correspondence: elena.campo@uah.es

† These authors contributed equally to this work.

**Abstract:** We present symbolic algorithms for computing the *g-asymptotes*, or *generalized asymptotes*, of a plane algebraic curve,  $\mathcal{C}$ , implicitly or parametrically defined. The *g-asymptotes* generalize the classical concept of asymptotes of a plane algebraic curve. Both notions have been previously studied for analyzing the geometry and topology of a curve at infinity points, as well as to detect the symmetries that can occur in coordinates far from the origin. Thus, based on this research, and in order to solve practical problems in the fields of science and engineering, we present the pseudocodes and implementations of algorithms based on the Puiseux series expansion to construct the *g-asymptotes* of a plane algebraic curve, implicitly or parametrically defined. Additionally, we propose some new symbolic methods and their corresponding implementations which improve the efficiency of the preceding. These new methods are based on the computation of limits and derivatives; they show higher computational performance, demanding fewer hardware resources and system requirements, as well as reducing computer overload. Finally, as a novelty in this research area, a comparative analysis for all the algorithms is carried out, considering the properties of the input curves and their outcomes, to analyze their efficiency and to establish comparative criteria between them.

**Keywords:** algebraic curves; infinity branches; convergent branches; approaching curves; generalized asymptotes; algorithm performance; symbolic computation



**Citation:** Campo-Montalvo, E.; Fernández de Sevilla, M.; Magdalena Benedicto, J.R.; Pérez-Díaz, S. Some New Symbolic Algorithms for the Computation of Generalized Asymptotes. *Symmetry* **2023**, *15*, 69. <https://doi.org/10.3390/sym15010069>

Academic Editors: Luca Grilli, Suleyman Şenyurt and Marian Ioan Munteanu

Received: 30 November 2022

Revised: 10 December 2022

Accepted: 21 December 2022

Published: 26 December 2022



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

This paper presents symbolic algorithms for determining the behavior at infinity of a plane algebraic curve,  $\mathcal{C}$ , by determining the *generalized asymptotes*, or *g-asymptotes*, of its branches at points with sufficiently large coordinates, i.e., at the infinity points. This concept generalizes the classical concept of (line) asymptote and its calculation methods (see [1,2]).

Both notions, *g-asymptotes* and *infinity branches*, are essential tools that allow us to analyze the geometry and topology of a curve at infinity points, as well as detect symmetries that can occur in coordinates far from the origin, which is applicable to solving practical problems in many fields of science and engineering. Thus, it should be noted that rational algebraic curves have been an area of research [3–6] that appears in many fields, such as, for example, ancient and modern architectural designs, in number theory problems [7,8], in biological forms [9], in error correction codes [10–12] in route planning for robots [13], and in cryptographic algorithms [7,14–17], among many other applications. In addition, they have recently acquired additional practical importance in the area of Computer-Aided Geometric Design for airplanes, automobiles, and household appliances, which would be unthinkable without computational help and the development of effective methods for the manipulation of algebraic curves and also surfaces [18–21]. The parameterizations of these

entities also play an important role in other aspects, such as line integral, representation, distribution of polynomial interpolation nodes [22], control theory [23], etc.

For this purpose, the pseudocodes of these algorithms are presented, as well as the corresponding implementations and a comparative analysis. We use the algebra software Maple.

To start with, Section 2 presents previous notions and explains the intuitive idea of *perfect curve* and *g-asymptote*. That is, given a curve  $C$ , a new curve  $\tilde{C}$  is said to be a *g-asymptote* of  $C$  if  $\tilde{C}$  is a curve of the smallest possible degree that approximates  $C$  at infinity (see [24]).

Based on these preliminaries, Section 3 presents the algorithms and pseudocodes which construct the generalized asymptotes from plane algebraic curves given by their implicit and parametric equations, considering the infinity branches that converge to the given curve. These pseudocodes are illustrated with examples that show the methods developed in this work, as well as the implementation programmed with the algebra system Maple, which are included in the Appendix C.

Thus, we present the pseudocodes and implementations of algorithms, based on previous studies that use the Puiseux series expansion to construct the *g-asymptotes* of a plane algebraic curve, implicitly or parametrically defined. Additionally, we propose some new symbolic methods and their corresponding implementations which improve the efficiency of the preceding research. These new methods are based on the computation of limits and derivatives, they show higher computational performance, demanding fewer hardware resources and system requirements, as well as reducing computer overload.

Accordingly, Section 4 presents, as a novelty in this research area, a comparative analysis for all the algorithms presented in this work, considering the properties of the input curves and their outcomes to analyze their efficiency and to establish comparative criteria between the algorithms.

For this purpose, we study several cases to observe the system overload, when Algorithm 1 or Algorithm 2 is executed (these two algorithms are based on the computation of the Puiseux series). However, for the parametric case, Algorithms 3 and 4 dramatically improve the efficiency of previous methods, since they are based on the computation of limits and derivatives. They show higher computational performance, requiring fewer hardware system resources available and a reduction in computer overload.

---

#### Algorithm 1 Computation of asymptotes of an implicit curve $C$

---

**Require:**  $C$ , plane algebraic curve defined by  $f(x, y) \in \mathbb{R}[x, y]$   
**Ensure:**  $\tilde{C}_i, i \leftarrow 1, \dots, k$  /\* Asymptotes of  $C$  \*/  
1:  $F(x, y, z) \leftarrow ProjectiveCurve(C)$   
2:  $P_1, \dots, P_m \leftarrow InfinityPoints(F(x, y, 0))$   
3:  $g(y, z) \leftarrow F(1, y, z)$   
4:  $\phi_1(z), \dots, \phi_k(z) \leftarrow PuiseuxSeries(g(y, z), z = 0, y)$   
5: **for all**  $\phi_i$  of  $P_j$  **do**  
6:      $r_i(z) \leftarrow z\phi_i(z^{-1})$   
7:      $B_i \leftarrow \{(z, r_i(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M_i\}$  /\* Definition 1 \*/  
8:      $\tilde{r}_i(z) \leftarrow m_i z + a_{1,i} z^{1-n_{1,i}/n_i} + \dots + a_{k,i,j} z^{1-n_{k,i,j}/n_i}$  /\* Equation (1) \*/  
9:      $n_i \leftarrow \deg(B_i)$   
10:      $\tilde{P}_i(t) \leftarrow (t^{n_i}, \tilde{r}_i(t^{n_i})) \in \mathbb{C}[t]^2$  /\* Equation (2) \*/  
11: **end for**  
12: **return**  $\tilde{C}_i \leftarrow \tilde{P}_i(t) \in \mathbb{C}[t]^2, i \leftarrow 1, \dots, k$

---

---

**Algorithm 2** Computation of asymptotes of a parametric curve  $\mathcal{C}$

---

**Require:**  $\mathcal{C}$ , plane algebraic curve defined by  
 $\mathcal{P}(s) \Leftarrow (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, p_i(s) = p_{i1}(s)/p_{i2}(s),$  /\* Equation (3) \*/  
 $\gcd(p_{i1}(s), p_{i2}(s)) = 1, i \leftarrow 1, 2$

**Ensure:**  $\tilde{\mathcal{C}}_i, i \leftarrow 1, \dots, k$  /\* Asymptotes of  $\mathcal{C}$  \*/

- 1:  $\mathcal{P}^*(s) \Leftarrow ProjectiveParametrization(\mathcal{P}(s))$  /\* Equation (4) \*/
- 2:  $P_1, \dots, P_m \Leftarrow InfinityPoints(\mathcal{P}^*(s))$
- 3:  $\ell_1(t), \dots, \ell_k(t) \Leftarrow PuiseuxSeries(p_{12}(s) - tp_{11}(s) = 0, t = 0, s)$
- 4: **for all**  $\ell_1(t)$  of  $P_j$  **do**
- 5:      $r_i(z) \Leftarrow p_2(\ell_i(z^{-1})) \in \mathbb{C} \ll t \gg, z \leftarrow t^{-1}$  /\* Remark 1 \*/
- 6:      $B_i \Leftarrow \{(z, r_i(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M_i\}$  /\* Definition 1 \*/
- 7:      $n_i \Leftarrow degree(B_i)$
- 8:      $\tilde{r}_i(z) \Leftarrow m_i z + a_{1,i} z^{1-n_{1,i}/n_i} + \dots + a_{k,i} z^{1-n_{k,i}/n_i}$  /\* Equation (1) \*/
- 9:      $\tilde{P}_i(t) \Leftarrow (t^{n_i}, \tilde{r}_i(t^{n_i})) \in \mathbb{C}[t]^2, i \leftarrow 1, \dots, k$  /\* Equation (2) \*/
- 10: **end for**
- 11: **return**  $\tilde{\mathcal{C}}_i \Leftarrow \tilde{P}_i(t) \in \mathbb{C}[t]^2, i \leftarrow 1, \dots, k$

---

**Algorithm 3** Computation of asymptotes of a parametric curve  $\mathcal{C}$  with limits

---

**Require:**  $\mathcal{C}$ , plane algebraic curve defined by  
 $\mathcal{P}(s) \Leftarrow (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, p_i(s) = p_{i1}(s)/p_{i2}(s),$  /\* Equation (3) \*/  
 $\gcd(p_{i1}(s), p_{i2}(s)) = 1, i \leftarrow 1, 2, \deg(p_{i1}) \leq \deg(p_{i2}), i = 1, 2.$

**Ensure:**  $\tilde{\mathcal{C}}_i, i \leftarrow 1, \dots, k$  /\* Asymptotes of  $\mathcal{C}$  \*/

- 1:  $\tau_1 \dots \tau_l \Leftarrow Roots(p_{12}(s)) \in \mathbb{C}$
- 2:  $n_{11} \dots n_{1l} \Leftarrow MultiplicityRoots(p_{12}(s))$
- 3:  $\rho_1 \dots \rho_k \Leftarrow Roots(p_{22}(s)) \in \mathbb{C}$
- 4:  $n_{21} \dots n_{2k} \Leftarrow MultiplicityRoots(p_{22}(s))$
- 5: **for all**  $\tau_i \in \{\tau_1 \dots \tau_l\}$  **do**
- 6:     **if**  $p_{22}(\tau_i) \neq 0$  **then**
- 7:          $\tilde{P}_i(t) \Leftarrow (t, p_{22}(\tau_i)) \in \mathbb{C}[t]^2$  /\* Horizontal Asymp. Corollary 1 \*/
- 8:     **else if**  $n_{2i} \geq 1$  **then**
- 9:          $f_{n_{2i}}(s) \Leftarrow \frac{p_2(s)}{p_1(s)^{n_{2i}/n_{1i}}}$
- 10:          $a_{n_{2i}} \Leftarrow \lim_{s \rightarrow \tau_i} f_{n_{2i}}(s)$
- 11:         **for**  $j \leftarrow n_{2i} - 1, \dots, 0$  **do**
- 12:              $f_j(s) \Leftarrow p_1(s)^{1/n_{1i}}(f_{j+1}(s) - a_{j+1})$
- 13:              $a_j \Leftarrow \lim_{s \rightarrow \tau_i} f_j(s)$  /\* Coeff. of the output parametrization. Theorem 1 \*/
- 14:         **end for**
- 15:          $\tilde{P}_i(t) \Leftarrow (t^{n_{1i}}, a_{n_{2i}} t^{n_{2i}} + a_j t^j + \dots + a_0 t^0) \in \mathbb{C}[t]^2$
- 16:     **end if**
- 17: **end for**
- 18: **for all**  $\rho_j \in \{\rho_1 \dots \rho_k\}$  &  $\rho_j \notin Roots(p_{12}(s))$  **do**
- 19:     **if**  $\exists \rho_j, p_{12}(\rho_j) \neq 0$  **then**
- 20:          $\tilde{P}_i(t) \Leftarrow (p_{12}(\rho_j), t) \in \mathbb{C}[t]^2$
- 21:          $i \leftarrow i + 1$  /\* Vertical Asymp. Corollary 1 \*/
- 22:     **end if**
- 23: **end for**
- 24: **return**  $\tilde{\mathcal{C}}_i \Leftarrow \tilde{P}_i(t) \in \mathbb{C}[t]^2, i \leftarrow 1, \dots, k$

---

In Section 5, a comparative analysis of the previous algorithms is carried out. For this purpose, some properties of the input curves are considered, as well as their results when constructing the respective g-asymptotes of the input curves. Then, the efficiency of each algorithm is analyzed, and a comparative criteria regarding the time of use of the CPU from the simplest curve to the curve with the highest complexity is established.

---

**Algorithm 4** Computation of asymptotes of a parametric curve  $\mathcal{C}$  with derivatives

---

**Require:**  $\mathcal{C}$  plane algebraic curve defined by  
 $\mathcal{P}(s) \leftarrow (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, p_i(s) = q_i(s)/q(s),$  /\* Equation (6) \*/  
 $\gcd(q_1(s), q_2(s), q(s)) = 1, i \leftarrow 1, 2.$

**Ensure:**  $\tilde{\mathcal{C}}_i, i \leftarrow 1, \dots, k$  /\* Asymptotes of  $\mathcal{C}^*$  \*/

- 1:  $\tau_1 \dots \tau_k \leftarrow \text{Roots}(q(s)) \in \mathbb{C}$
- 2:  $n_1 \dots n_k \leftarrow \text{MultiplicityRoots}(q(s))$
- 3:  $\sigma(s) \leftarrow \frac{q_2(s)}{q_1(s)}$  /\* Theorem 2 \*/
- 4: **for all**  $\tau_i \in \{\tau_1 \dots \tau_k\}$  **do**
- 5:  $\bar{q}_i(s) \leftarrow \frac{q(s)}{(s - \tau_i)^{n_i}}$
- 6:  $\rho_i(s) \leftarrow \left(\frac{q_1(s)}{\bar{q}_i}\right)^{\frac{1}{n_i}}$
- 7:  $a_{n_i} \leftarrow \sigma(\tau_i)$
- 8: **for**  $j \leftarrow (n_i - 1), \dots, 0$  **do**
- 9:  $a_j = \frac{1}{(n_j - j)!} \cdot \frac{\partial^{n_i-1-j} \sigma(s)}{\partial s^{n_i-1-j}} \cdot \rho_i^{(n_i-j)}(\tau_i)$  /\* Coeff. of the Parametrization \*/
- 10: **end for**
- 11:  $\tilde{\mathcal{P}}_i(t) \leftarrow (t^{n_i}, a_{n_i}t^{n_i} + \dots + a_0t^0) \in \mathbb{C}[t]^2$
- 12: **end for**
- 13: **return**  $\tilde{\mathcal{C}}_i \leftarrow \tilde{\mathcal{P}}_i(t) \in \mathbb{C}[t]^2, i \leftarrow 1, \dots, k$

---

Finally, the conclusions and lines of future work are presented. More precisely, we propose to generalize the study to curves defined by a nonrational parametrization, as well as to surfaces, and study the families of existing  $g$ -asymptotes. Moreover, we advance the idea that these methods could be extended to improve fuzzy-based algorithms.

We should point out that the novelties concerning previous works are based on the presentation of the symbolic algorithms and the pseudocode with the algebra system Maple to construct the asymptotes of a plane algebraic curve.

**2. Preliminaries and Terminology**

This section starts by introducing the notions of *infinity branches*, *convergent branches*, and *approaching curves*, derived from previous research (see [24,25]).

Let  $\mathcal{C}$  be an irreducible plane algebraic curve defined in the affine space by an irreducible polynomial  $f(x, y) \in \mathbb{R}[x, y]$ . Due to practical implications, the curve is assumed to be real and, therefore, the implicit polynomial is defined over  $\mathbb{R}$ . Let  $\mathcal{C}^*$  be its corresponding projective curve defined by the homogeneous polynomial

$$F(x, y, z) = f_d(x, y) + zf_{d-1}(x, y) + z^2f_{d-2}(x, y) + \dots + z^d f_0(x, y) \in \mathbb{R}[x, y, z],$$

with  $d := \deg(\mathcal{C})$ , and let  $(1 : m : 0), m \in \mathbb{C}$  be the form of the points of infinity of  $\mathcal{C}^*$  (if the point of infinity  $(0 : 1 : 0)$  exists, a linear change in coordinates must be applied).

Under these conditions, starting from the curve defined by the polynomial  $g(y, z) = F(1 : y : z)$  and calculating the Puiseux series,  $\varphi_i, i = 1 \dots \deg_y(g)$  from  $g(y, z) = 0$  around  $z = 0$ , we obtain the *branches* of  $\mathcal{C}$  (see [24]). In the following, we denote as

$$\varphi(t) = m + a_1t^{N_1/N} + a_2t^{N_2/N} + a_3t^{N_3/N} + \dots, \quad a_i \neq 0, \forall i \in \mathbb{N},$$

with  $N_i \in \mathbb{N}, i = 1, \dots,$  and  $0 < N_1 < N_2 < \dots$  one of these series. Therefore,  $g(\varphi(t), t) = 0$  in a neighborhood of  $t = 0$  where  $\varphi(t)$  converges.

**Definition 1.** An infinity branch of a plane algebraic curve  $\mathcal{C}$ , at the infinity point  $P = (1 : m : 0)$ ,  $m \in \mathbb{C}$ , is the set  $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M\}$ , where  $r(z) = z\varphi_i(z^{-1}) = mz + a_1z^{1-N_1/N} + a_2z^{1-N_2/N} + a_3z^{1-N_3/N} + \dots$  and  $M$  is some natural number.

**Definition 2.** Given two branches,  $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M\}$  and  $\bar{B} = \{(z, \bar{r}(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > \bar{M}\}$ , we say that they are convergent if  $\lim_{z \rightarrow \infty} (\bar{r}(z) - r(z)) = 0$ .

**Definition 3.** Let  $\mathcal{C}$  be a plane algebraic curve with an infinity branch  $B$ . We say that a curve  $\mathcal{D}$  approaches  $\mathcal{C}$  at the branch  $B$ , if  $\lim_{z \rightarrow \infty} d((z, r(z)), \mathcal{D}) = 0$ , i.e.,  $d(p, \mathcal{D}) = \min\{d(p, q) : q \in \mathcal{D}\}$ .

In [24], we show that if  $\mathcal{C}$  is a plane curve with an infinity branch  $B$ , then a plane curve  $\bar{\mathcal{C}}$  approximates  $\mathcal{C}$  into  $B$ , if and only if  $\bar{\mathcal{C}}$  has an infinity branch  $\bar{B}$  such that  $B$  and  $\bar{B}$  are convergent.

From the previous concepts, the definitions of perfect curve and generalized asymptote are obtained (see [24]).

**Definition 4.** A curve of degree  $d$  is a perfect curve if it cannot be approximated by any curve of degree less than  $d$ .

Note that a curve that is not perfect can be approximated by other curves of a lesser degree.

**Definition 5.** Let  $\mathcal{C}$  be a plane algebraic curve with an infinity branch  $B$ . A curve,  $\tilde{\mathcal{C}}$ , is a  $g$ -asymptote or generalized asymptote of  $\mathcal{C}$  in  $B$ , if it is a perfect curve that approximates  $\mathcal{C}$  in  $B$ .

Let  $\mathcal{C}$  be a curve with a branch  $B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M\}$ , where  $r(z) = mz + a_1z^{1-N_1/N} + \dots + a_kz^{1-N_k/N} + a_{k+1}z^{1-N_{k+1}/N} + \dots$ , with coefficients  $a_1, a_2, \dots \in \mathbb{C} \setminus \{0\}$ ,  $m \in \mathbb{C}$ ,  $N, N_1, N_2 \dots \in \mathbb{N}$ , and  $0 < N_1 < N_2 < \dots$ .

Suppose that  $N_k \leq N < N_{k+1}$ , i.e., the terms  $a_jz^{1-N_j/N}$  with  $j \geq k + 1$  have a negative exponent. In the following, we write

$$r(z) = mz + a_1z^{1-n_1/n} + \dots + a_kz^{1-n_k/n} + a_{k+1}z^{1-N_{k+1}/N} + \dots$$

with  $\gcd(N, N_1, \dots, N_k) = b, N_j = n_jb, N = nb, j = 1, \dots, k$ . That is, we simplify the exponents such that  $\gcd(n, n_1, \dots, n_k) = 1$ . Note that  $0 < n_1 < n_2 < \dots, n_k \leq n$ , and  $N < n_{k+1}$ , i.e., the terms  $a_jz^{1-N_j/N}$  with  $j \geq k + 1$  have negative exponents. The terms with non-negative exponent of  $r(z)$  are

$$\tilde{r}(z) = mz + a_1z^{1-n_1/n} + \dots + a_kz^{1-n_k/n}. \tag{1}$$

Applying the change  $z = t^n$ , we obtain a proper parametrization of a curve  $\tilde{\mathcal{C}}$

$$\tilde{\mathcal{P}}(t) = (t^n, mt^n + a_1t^{n-n_1} + \dots + a_kt^{n-n_k}) \in \mathbb{C}[t]^2, \tag{2}$$

where  $n, n_1, \dots, n_k \in \mathbb{N}, \gcd(n, n_1, \dots, n_k) = 1$ , and  $0 < n_1 < \dots < n_k$ , which is an asymptote of  $\mathcal{C}$  (see [24]).

### 3. Symbolic Algorithms for the Computation of Generalized Asymptotes

In this section, we describe four algorithms that construct the parametrizations of the generalized asymptotes of the infinity branches of a curve  $\mathcal{C}$ , all of them are described by corresponding pseudocode. Furthermore, we study the behavior and execution of the algorithms.

The next Subsection introduces Algorithm 1 which computes the  $g$ -asymptotes of a curve given by its implicit expression. Section 3.2, Algorithm 2 is applied to a parametrically given curve. Finally, the last two subsections show the improvement of Algorithm 2,

with two new algorithms, Algorithms 3 and 4, that dramatically improve the previous results for the parametric case.

### 3.1. Algorithm for Curves Implicitly Defined

The following algorithm computes the parametrizations of the asymptotes of the infinity branches of the curve  $C$ , which is implicitly defined. It should be noted that the below algorithm was implemented with the mathematical software Maple (see Appendix C).

**Example 1.** Let  $C$  be a curve of degree  $d = 6$ , defined by the irreducible polynomial  $f(x, y) = y^6 + 2y^5x + 3x^2 + 4xy \in \mathbb{R}[x, y]$ . The points of infinity are  $P_1 = (1 : 0 : 0)$  and  $P_2 = (1 : -2 : 0)$ .

**Iteration 1:** Let  $P_1 = (1 : 0 : 0)$ .

We have the branch  $B_1 = \{(z, r_1(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M_1\}$ , with

$$r_1(z) = -\frac{48^{1/5}}{2}z - \frac{-72^{1/5}}{12}z^{-3} + \frac{108^{1/5}}{18}z^{-7} - \frac{-162^{1/5}13}{432}z^{-11} + \dots$$

$$(a) \tilde{r}_1(z) = -\frac{48^{1/5}}{2}z. \quad (b) \tilde{\mathcal{P}}_1(t) = \left(t^5, -\frac{48^{1/5}}{2}t\right).$$

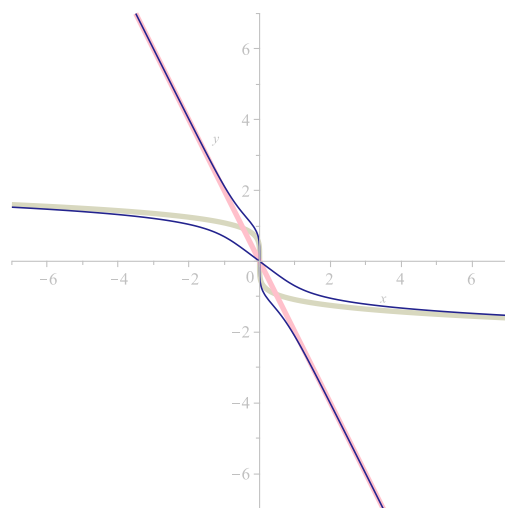
**Iteration 2:** Let  $P_2 = (1 : -2 : 0)$ .

We have the branch  $B_2 = \{(z, r_2(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M_2\}$  with

$$r_2(z) = -2z - \frac{5}{32}z^{-3} + \dots$$

$$(a) \tilde{r}_2(z) = -2z. \quad (b) \tilde{\mathcal{P}}_2(t) = (t, -2t).$$

Figure 1 represents the curve  $C$  and its generalized asymptotes  $\tilde{C}_1$  and  $\tilde{C}_2$ , defined by the parametrizations  $\tilde{\mathcal{P}}_1(t)$  and  $\tilde{\mathcal{P}}_2(t)$ , respectively.



**Figure 1.** Infinity asymptotes  $\tilde{C}_1$  (beige) and  $\tilde{C}_2$  (pink) of the curve  $C$ .

### 3.2. Algorithm for Curves Parametrically Defined

Throughout this paper so far, we have dealt with algebraic plane curves implicitly defined. Now, we present a method to compute infinity branches and g-asymptotes of a plane curve from their parametric representation, without implicitizing. This method also involves the computation of Puiseux series and infinity branches (see [26]).

Let  $C$  be a plane curve defined by the parametrization

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, p_i(s) = p_{i1}(s)/p_{i2}(s), \quad \gcd(p_{i1}, p_{i2}) = 1, \quad i = 1, 2. \quad (3)$$

If  $C^*$  represents the projective curve associated to  $C$ , we have that a parametrization of  $C^*$  is given by  $\mathcal{P}^*(s) = (p_1(s) : p_2(s) : 1)$  or, equivalently,

$$\mathcal{P}^*(s) = (1 : p_2(s)/p_1(s) : 1/p_1(s)). \tag{4}$$

We assume that we prepared the input curve  $C$  through a suitable linear change in coordinates (if necessary), such that  $(0 : 1 : 0)$  is not a point at infinity of  $C^*$ .

In order to compute the g-asymptotes of  $C$ , first we need to determine the infinity branches of  $C$ . That is, the sets

$$B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C}, \|z\| > M\}, \text{ where } r(z) = z\varphi(z^{-1}).$$

Taking into account Definition 1, we have that  $f(z, r(z)) = F(1 : \varphi(z^{-1}) : z^{-1}) = F(1 : \varphi(t) : t) = 0$  around  $t = 0$ , where  $t = z^{-1}$  and  $F$  is the polynomial defining implicitly  $C^*$ . Observe that, in this section, we are given the parametrization  $\mathcal{P}^*(s)$  of  $C^*$  and then,  $F(\mathcal{P}^*(s)) = F(1 : p_2(s)/p_1(s) : 1/p_1(s)) = 0$ . Thus, intuitively speaking, in order to compute the infinity branches of  $C$ , and in particular the series  $\varphi$ , one needs to rewrite the parametrization  $\mathcal{P}^*(s)$  in the form  $(1 : \varphi(t) : t)$  around  $t = 0$ . For this purpose, the idea is to look for a value of the parameter  $s$ , say  $\ell(t) \in \mathbb{C} \ll t \gg$ , such that  $\mathcal{P}^*(\ell(t)) = (1 : \varphi(t) : t)$  around  $t = 0$ .

Hence, from the above reasoning, we deduce that, first, we have to consider the equation  $1/p_1(s) = t$ , or equivalently  $p_{12}(s) - tp_{11}(s) = 0$ , and we solve it in the variable  $s$  around  $t = 0$ .

From Puiseux’s Theorem, there exist solutions  $\ell_1(t), \ell_2(t), \dots, \ell_k(t) \in \mathbb{C} \ll t \gg$  such that,  $p_{12}(\ell_i(t)) - tp_{11}(\ell_i(t)) = 0, i \in \{1, \dots, k\}$ , in a neighborhood of  $t = 0$ .

Thus, for each  $i \in \{1, \dots, k\}$ , there exists  $M_i \in \mathbb{R}^+$  such that the points  $(1 : \varphi_i(t) : t)$  or equivalently, the points  $(t^{-1} : t^{-1}\varphi_i(t) : 1)$ , where

$$\varphi_i(t) = p_2(\ell_i(t))/p_1(\ell_i(t)), \tag{5}$$

are in  $C^*$  for  $\|t\| < M_i$  (note that  $\mathcal{P}^*(\ell(t)) \in C^*$ , since  $\mathcal{P}^*(\ell(t))$  is a parametrization of  $C^*$ ). Observe that  $\varphi_i(t)$  is a Puiseux series, since  $p_2(\ell_i(t))$  and  $p_1(\ell_i(t))$  can be written as Puiseux series and  $\mathbb{C} \ll t \gg$  is a field.

Finally, we set  $z = t^{-1}$ . Then, we have that the points  $(z, r_i(z))$ , where  $r_i(z) = z\varphi_i(z^{-1})$  are in  $C$  for  $\|z\| > M_i^{-1}$ . Hence, the infinity branches of  $C$  are the sets  $B_i$ , such that  $B_i = \{(z, r_i(z)) \in \mathbb{C}^3 : z \in \mathbb{C}, \|z\| > M_i^{-1}\}, i \in \{1, \dots, k\}$ .

**Remark 1.** Note that the series  $\ell_i(t)$  satisfies that  $p_1(\ell_i(t))t = 1$ , for  $i \in \{1, \dots, k\}$ . Then, from equality (5), we have that  $\varphi_i(t) = \frac{p_2(\ell_i(t))}{p_1(\ell_i(t))} = p_2(\ell_i(t))t$ , and  $r_i(z) = z\varphi_i(z^{-1}) = p_2(\ell_i(z^{-1}))$ .

Once we have the infinity branches, we can compute a g-asymptote for each of them by simply removing the terms with negative exponent from  $r_i(z)$ .

The following algorithm computes the infinity branches of a given parametric curve and provides a g-asymptote for each of them. We also illustrate the algorithm with an example. The algorithm is implemented in Maple (see Appendix C).

**Example 2.** Let  $C$  be the plane curve defined by

$$\mathcal{P}(s) = \left( \frac{s^2 + 5}{s(s-1)(s-2)^2}, \frac{s^2 + 3s + 1}{s(s-2)} \right) \in \mathbb{R}(s)^2.$$

We apply Algorithm 2 to compute the asymptotes of  $C$ . We start by computing the Puiseux solutions of the equation  $p_{12}(s) - tp_{11}(s) = 0$  around  $t = 0$ . We obtain:

$$\ell_1(t) = -\frac{3475}{256}t^3 + \frac{25}{8}t^2 - \frac{5}{4}t + \dots$$



$$\ell_2(t) = \frac{375800710285\sqrt{2}}{254803968}t^{7/2} - \frac{233069}{512}t^3 + \frac{32580935\sqrt{2}}{442368}t^{5/2} - \frac{409}{16}t^2 + \frac{1909\sqrt{2}}{384}t^{3/2} - \frac{19}{8}t + \dots$$

$$\ell_3(t) = 924t^3 + 48t^2 + 6t + 1 + \dots$$

**Iteration 1:** Let  $\ell_1(t) = -\frac{3475}{256}t^3 + \frac{25}{8}t^2 - \frac{5}{4}t + \dots$

$$r_1(z) = \frac{2}{5}z - \frac{3}{4} - \frac{1}{8}z^{-1} - \frac{5335}{256}z^{-2} + \dots$$

$$(a) \tilde{r}_1(z) = \frac{2}{5}z - \frac{3}{4}. \quad (b) \tilde{\mathcal{P}}_1(t) = \left(t, \frac{2}{5}t - \frac{3}{4}\right).$$

**Iteration 2:**

$$\text{Let } \ell_2(t) = -\frac{233069}{512}t^3 + \frac{32580935\sqrt{2}}{442368}t^{5/2} - \frac{409}{16}t^2 + \frac{1909\sqrt{2}}{384}t^{3/2} - \frac{19}{8}t + \dots$$

$$r_2(z) = \frac{11\sqrt{2}\sqrt{z}}{6} + \frac{263}{72} - \frac{12409\sqrt{2}}{3456\sqrt{z}} + \frac{241}{16z} - \frac{155680499\sqrt{2}}{3981312z^{3/2}} + \frac{116201}{512z^2} + \dots$$

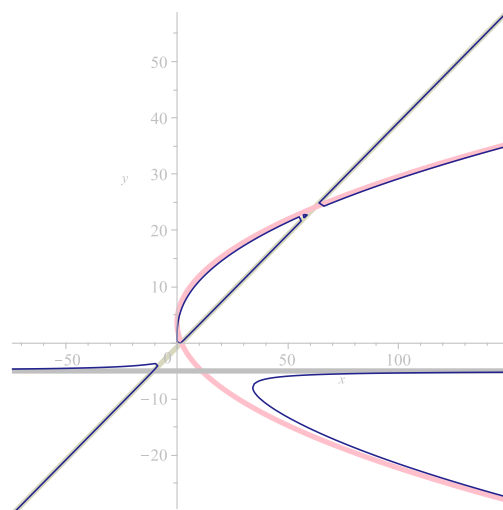
$$(a) \tilde{r}_2(z) = \frac{11\sqrt{2}\sqrt{z}}{6} + \frac{263}{72}. \quad (b) \tilde{\mathcal{P}}_2(t) = \left(t^2, \frac{11\sqrt{2}t}{6} + \frac{263}{72}\right).$$

**Iteration 3:** Let  $\ell_3(t) = 924t^3 + 48t^2 + 6t + 1 + \dots$

$$r_3(z) = -5 - 30z^{-1} - 456z^{-2} - 9156z^{-3} - \dots$$

$$(a) \tilde{r}_3(z) = -5. \quad (b) \tilde{\mathcal{P}}_3(t) = (t, -5).$$

The obtained asymptotes are the curves  $\tilde{\mathcal{C}}_1$ ,  $\tilde{\mathcal{C}}_2$ , and  $\tilde{\mathcal{C}}_3$  defined by the proper parametrizations,  $\tilde{\mathcal{P}}_1(t)$ ,  $\tilde{\mathcal{P}}_2(t)$  and  $\tilde{\mathcal{P}}_3(t)$  (see Figure 2).



**Figure 2.** Infinity asymptotes  $\tilde{\mathcal{C}}_1$  (beige),  $\tilde{\mathcal{C}}_2$  (pink), and  $\tilde{\mathcal{C}}_3$  (gray) of the curve  $\mathcal{C}$ .

### 3.3. Algorithm with Limits for Curves Parametrically Defined

As described previously, generalized asymptotes are a fundamental tool to analyze the behavior of a curve at infinity. In the two previous sections, two algorithmic solutions were presented, one for the case of implicitly defined planar algebraic curves and another for the case of parametric curves.

In both cases, the calculation of the g-asymptotes was carried out from the calculation of the infinity branches of the given curves by means of Puiseux series. Algorithm 1



calculates the g-asymptotes of an implicitly defined curve, while Algorithm 2 constructs these generalized asymptotes from the equation of the curve expressed in parametric way.

This subsection presents a method that improves the results of the previously mentioned algorithms, providing a more efficient computational solution, which can be applied to the case of plane algebraic curves expressed parametrically (see Algorithm 3). Furthermore, although this solution is developed for the case of plane algebraic curves, it can be trivially generalized to the case of rational curves defined in  $n$ -dimensional space. The idea of this generalization can be extended in more detail to the case of not necessarily rational curves.

For this purpose, we consider a plane rational curve,  $\mathcal{C}$ , defined by a parameterization according to Equation (7):

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \quad p_i(s) = p_{i1}(s)/p_{i2}(s), \quad \gcd(p_{i1}(s), p_{i2}(s)) = 1, \quad i = 1, 2.$$

In the following, it is assumed without loss of generality that  $\deg(p_{i1}(s)) \leq \deg(p_{i2}(s))$ ,  $\deg(p_{i2}(s)) = d_i$ ,  $i = 1, 2$  (otherwise, a linear change in variables is applied). With this,  $\lim_{s \rightarrow \infty} p_i(s) \neq \infty$ ,  $i = 1, 2$ , and the infinity branches of  $\mathcal{C}$  will be represented when the variable  $s$  moves around the different roots of the denominators  $p_{12}(s)$  and  $p_{22}(s)$ .

The following theorem shows how to obtain a g-asymptote for each of these infinity branches, by calculating the simple limits of rational functions built from  $\mathcal{P}(s)$ .

**Theorem 1.** *Let  $\mathcal{C}$  be a curve defined by a parameterization (see Equation (7))*

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \quad p_i(s) = p_{i1}(s)/p_{i2}(s), \quad \gcd(p_{i1}(s), p_{i2}(s)) = 1, \quad i = 1, 2,$$

where  $\deg(p_{i1}(s)) \leq \deg(p_{i2}(s)) = d_i$ ,  $i = 1, 2$ . Let  $\tau \in \mathbb{C}$  be such that  $p_{i2}(s) = (s - \tau)^{n_i} \bar{p}_{i2}(s)$  where  $\bar{p}_{i2}(\tau) \neq 0$ ,  $i = 1, 2$ , and  $n_i \geq 1$ , and let  $B$  be the corresponding infinity branch. A g-asymptote of the curve  $\mathcal{C}$  on the branch  $B$  is defined by the parameterization

$$\tilde{\mathcal{P}}(t) = (t^{n_1}, a_{n_2}t^{n_2} + a_{n_2-1}t^{n_2-1} + \dots + a_0),$$

where

$$\begin{aligned} a_{n_2} &= \lim_{s \rightarrow \tau_i} f_{n_2}(s), & f_{n_2}(s) &= \frac{p_2(s)}{p_1(s)^{n_2/n_1}} \\ a_{n_2-1} &= \lim_{s \rightarrow \tau_i} f_{n_2-1}(s), & f_{n_2-1}(s) &= p_1(s)^{1/n_1} (f_{n_2}(s) - a_{n_2}) \\ &\vdots & &\vdots \\ a_{n_2-i} &= \lim_{s \rightarrow \tau_i} f_{n_2-i}(s), & f_{n_2-i}(s) &= p_1(s)^{1/n_1} (f_{n_2-i+1}(s) - a_{n_2-i+1}) \\ &\vdots & &\vdots \\ a_0 &= \lim_{s \rightarrow \tau_i} f_0(s), & f_0(s) &= p_1(s)^{1/n_1} (f_1(s) - a_1) \end{aligned}$$

**Remark 2.** *From Theorem 1 we obtain the parameterization*

$$\tilde{\mathcal{P}}(t) = (t^{n_1}, a_{n_2}t^{n_2} + a_{n_2-1}t^{n_2-1} + \dots + a_0) \text{ and } n_1 \geq n_2$$

(otherwise,  $(0 : 1 : 0)$  would be an infinity point on the input curve).

Note that the degree of the resulting curve is not necessarily  $n_1$ , since  $\tilde{\mathcal{P}}(t)$  could be an improper parameterization, which is equivalent to  $\gcd(n_1, n_2, \dots, n_2 - j) \neq 0$  for each  $j = 0, \dots, n_2 - 1$ , such that  $a_{n_2-j} \neq 0$ . Under these conditions, let  $\gcd(n_1, n_2, \dots, n_2 - j) = \beta$  for each  $j = 0, \dots, n_2 - 1$ , such that  $a_{n_2-j} \neq 0$ . So, if  $n := n_1 / \beta$ , then

$$\mathcal{M}(t) = \tilde{\mathcal{P}}(t^{1/\beta}) = (t^n, a_{n_2}t^{n_2/\beta} + a_{n_2-1}t^{(n_2-1)/\beta} + \dots + a_0) \in \mathbb{K}[t]^2,$$

a proper reparametrization of  $\tilde{\mathcal{P}}(t)$  (see ([27] Section 3)).

The following corollary discusses the special case of vertical and horizontal g-asymptotes, that is, lines of the form  $x - a$  or  $y - b$ , where  $a, b \in \mathbb{C}$ . More precisely, it is proved that

these asymptotes are obtained from the noncommon roots of the denominators of a given parameterization. It should be noted that, in practical engineering design and application modeling, rational curves are normally represented by numerical coefficients and, in general,  $\mathcal{P}(s)$  satisfies that  $\gcd(p_{12}(s), p_{22}(s)) = 1$ , which implies that the mentioned case usually arises in practical problems.

**Corollary 1.** *Let  $\mathcal{C}$  be a curve defined by a parameterization (see Equation (7))*

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \quad p_i(s) = p_{i1}(s)/p_{i2}(s), \quad \gcd(p_{i1}(s), p_{i2}(s)) = 1,$$

where  $\deg(p_{i1}(s)) \leq \deg(p_{i2}(s)), i = 1, 2$ .

1. Let  $\tau \in \mathbb{C}$  be such that  $p_{12}(t) = (t - \tau)^{n_1} \bar{p}_{12}(t)$  where  $p_{22}(\tau)\bar{p}_{12}(\tau) \neq 0$  and  $n_1 \geq 1$ . An asymptote of  $\mathcal{C}$ , corresponding to the point of infinity  $(1 : 0 : 0)$ , is the horizontal line  $y - p_2(\tau) = 0$  defined by the parameterization  $\tilde{\mathcal{P}}(t) = (t, p_2(\tau))$ .
2. Let  $\tau \in \mathbb{C}$  be such that  $p_{22}(t) = (t - \tau)^{n_2} \bar{p}_{22}(t)$  where  $p_{12}(\tau)\bar{p}_{22}(\tau) \neq 0$  and  $n_2 \geq 1$ . An asymptote of  $\mathcal{C}$ , corresponding to the point of infinity  $(0 : 1 : 0)$ , is the vertical line  $x - p_1(\tau) = 0$  defined by the parameterization  $\tilde{\mathcal{P}}(t) = (p_1(\tau), t)$ .

As stated previously, Algorithm 3 improves the results of the methods described above, especially with regard to Algorithm 2, showing a clear improvement in efficiency and reducing the amount of hardware resources required of the computer when constructing the asymptotes of a parametric curve using limits (instead of using the Puiseux series expansion). Algorithm 3 is implemented in Maple (see Appendix C).

**Example 3.** *Let  $\mathcal{C}$  be the plane algebraic curve defined by the parameterization*

$$\mathcal{P}(s) = \left( \frac{4(s^2 + 1)}{s^2(s - 2)^3}, \frac{2s^2 + 2s - 1}{s^2(s - 2)^2} \right) \in \mathbb{R}(s)^2.$$

We apply Algorithm 3. Then, we compute the roots of  $p_{12}(s)$  which are  $\tau_1 = 0$  and  $\tau_2 = 2$ , with multiplicities  $n_{11} = 2$  and  $n_{12} = 3$ , respectively. The multiplicities of these roots in  $p_{22}(s)$  are  $n_{21} = 2$  and  $n_{22} = 2$ .

**Root  $\tau_1 = 0$ :** *It is obtained that:*

- There is no horizontal asymptote.
- The parameterization coefficients, considering the multiplicities  $n_{11} = 2$  and  $n_{21} = 2$ , are:

$$\begin{aligned} a_2 &= \lim_{s \rightarrow 0} f_2(s) = \frac{1}{2}, & f_2(s) &= \frac{p_2(s)}{p_1(s)}. \\ a_1 &= \lim_{s \rightarrow 0} f_1(s) = \frac{5\sqrt{2}i}{8}, & f_1(s) &= p_1(s)^{1/2}(f_2(s) - a_2). \\ a_0 &= \lim_{s \rightarrow 0} f_0(s) = \frac{31}{32}, & f_0(s) &= p_1(s)^{1/2}(f_1(s) - a_1). \end{aligned}$$

- The parameterization  $\tilde{\mathcal{P}}_1(t) = \left( t^2, \frac{1}{2}t^2 + \frac{5\sqrt{2}i}{8}t + \frac{31}{32} \right)$  is proper (see Remark 2).

**Root  $\tau_2 = 2$ :** *It is obtained that:*

- There is no horizontal asymptote.
- The coefficients of the proper parameterization, considering the multiplicities  $n_{12} = 3$  and  $n_{22} = 2$ , are:

$$\begin{aligned} a_2 &= \lim_{s \rightarrow 2} f_2(s) = \frac{11 \cdot 5^{1/3}}{20}, & f_2(s) &= \frac{p_2(s)}{p_1(s)^{2/3}}. \\ a_1 &= \lim_{s \rightarrow 2} f_1(s) = \frac{7 \cdot 5^{2/3}}{300}, & f_1(s) &= p_1(s)^{1/3}(f_2(s) - a_2). \\ a_0 &= \lim_{s \rightarrow 2} f_0(s) = \frac{77}{400}, & f_0(s) &= p_1(s)^{1/3}(f_1(s) - a_1). \end{aligned}$$

- The parameterization  $\tilde{\mathcal{P}}_2(t) = \left( t^3, \frac{11 \cdot 5^{1/3}}{20} t^2 + \frac{7 \cdot 5^{2/3}}{300} t + \frac{77}{400} \right)$  is proper (see Remark 2).

Finally, we observe that there are no vertical asymptotes. Consequently, the asymptotes are the curves  $\tilde{\mathcal{C}}_1$  and  $\tilde{\mathcal{C}}_2$  defined by the proper parameterizations:

$$\tilde{\mathcal{P}}_1(t) = \left( t^2, \frac{1}{2} t^2 + \frac{5\sqrt{2}i}{8} t + \frac{31}{32} \right), \quad \tilde{\mathcal{P}}_2(t) = \left( t^3, \frac{11 \cdot 5^{1/3}}{20} t^2 + \frac{7 \cdot 5^{2/3}}{300} t + \frac{77}{400} \right).$$

Figure 3 plots the curve  $\mathcal{C}$  and the asymptotes  $\tilde{\mathcal{C}}_1$  and  $\tilde{\mathcal{C}}_2$ .

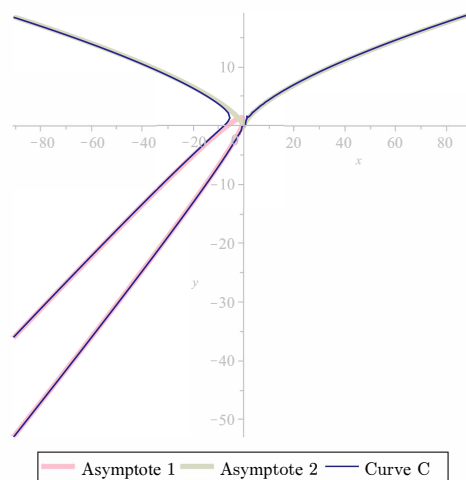


Figure 3. Asymptotes  $\tilde{\mathcal{C}}_1$  and  $\tilde{\mathcal{C}}_2$  of the curve  $\mathcal{C}$  calculated with limits.

### 3.4. Algorithm with Derivatives for Curves Parametrically Defined

In this subsection, a method is developed that allows calculating all the g-asymptotes and branches of a plane algebraic curve defined by a parameterization by determining some derivatives of univariate functions constructed from the input parameterization. For this purpose, we consider the plane algebraic curve,  $\mathcal{C}$ , defined by the parameterization,

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \quad p_i(s) = q_i(s)/q(s), \quad \gcd(q_1(s), q_2(s), q(s)) = 1, \quad i = 1, 2. \tag{6}$$

It is assumed that all roots of the denominators can be written as  $\tau \in \mathbb{C}$ , such that  $q(s) = (s - \tau)^n \bar{q}(s)$ , where  $\bar{q}(\tau)q_1(\tau) \neq 0$  and  $n \geq 1$ . Otherwise, it would be considered a linear change in coordinates, which would have to be undone later.

**Theorem 2.** Let  $\mathcal{C}$  be a curve defined by the parameterization

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \quad p_i(s) = q_i(s)/q(s), \quad \gcd(q_1(s), q_2(s), q(s)) = 1, \quad i = 1, 2.$$

Let  $\tau \in \mathbb{C}$  be such that  $q(s) = (s - \tau)^n \bar{q}(s)$ , where  $\bar{q}(\tau)q_1(\tau) \neq 0$  and  $n \geq 1$ . Let

$$\sigma(s) := \frac{q_2(s)}{q_1(s)} \quad \text{y} \quad \rho(s) := \left( \frac{q_1(s)}{\bar{q}(s)} \right)^{1/n}.$$

Let  $B$  be the corresponding infinity branch. A g-asymptote of  $B$  is defined by the parameterization

$$\tilde{\mathcal{P}}(t) = (t^n, a_n t^n + a_{n-1} t^{n-1} + \dots + a_0),$$

where  $a_n = \sigma(\tau)$

and for  $0 \leq i \leq n - 1$ , 
$$a_i = \frac{1}{(n-i)!} \cdot \frac{\partial^{n-1-i}}{\partial s^{n-1-i}} \left( \frac{\partial \sigma}{\partial s}(s) \cdot \rho(s)^{(n-i)} \right) (\tau)$$

**Remark 3.** From Theorem 2, we obtain the parameterization

$$\tilde{\mathcal{P}}(t) = (t^n, a_n t^n + a_{n-1} t^{n-1} + \dots + a_0),$$

where  $a_n = \sigma(\tau)$  and for  $0 \leq i \leq n - 1$ ,

$$a_i = \frac{1}{(n - i)!} \cdot \frac{\partial^{n-1-i}}{\partial s^{n-1-i}} \left( \frac{\partial \sigma}{\partial s}(s) \cdot \rho(s)^{(n-i)} \right) (\tau).$$

Note that the degree of the resulting curve is not necessarily  $n$ , since  $\tilde{\mathcal{P}}(t)$  could be an improper parameterization, which is equivalent to  $\gcd(n, \dots, n - j) \neq 0$  for each  $j = 0, \dots, n$ , such that  $a_n \neq 0$ .

Therefore, we have that  $\gcd(n, \dots, n - j) = \beta$  for each  $j = 0, \dots, n$  such that  $a_{n-j} \neq 0$ . We have that  $\mathcal{M}(t) = \tilde{\mathcal{P}}(t^{1/\beta})$  is a proper reparameterization of  $\tilde{\mathcal{P}}(t)$  (see ([27] Section 3)).

Next, Algorithm 4 is introduced. It performs the calculation of the asymptotes of a plane algebraic curve, expressed in parametric form, applying the results of the Theorem 2. This algorithm is implemented in Map1e (see Appendix C).

**Example 4.** Let  $\mathcal{C}$  be the plane curve defined by the parameterization

$$\mathcal{P}(s) = \left( \frac{s^4 - s^3 + 5s^2 + 2s + 1}{s^4(s - 1)(s - 2)}, \frac{2s^4 - 3s^3 - 2s^2 - 26s - 18}{s^4(s - 1)(s - 2)} \right) \in \mathbb{R}(s)^2.$$

We compute the asymptotes of  $\mathcal{C}$  using Algorithm 4. For this purpose, it is first observed that  $p(s)$  has the roots  $\tau_1 = 0, \tau_2 = 1, \tau_3 = 2$ , with multiplicities  $n_1 = 4, n_2 = 1$  and  $n_3 = 1$ . Let

$$\sigma(s) = \frac{2s^4 - 3s^3 - 2s^2 - 26s - 18}{s^4 - s^3 + 5s^2 + 2s + 1}.$$

**Root  $\tau_1 = 0$ :** We have  $\rho_1(s) = \frac{((s^4 - s^3 + 5s^2 + 2s + 1)(s - 2)^3(s - 1)^3)^{1/4}}{(s - 2)(s - 1)}$ .

- The parameterization coefficients are calculated, knowing that the multiplicity of  $\tau_1$  is  $n_1 = 4$ .

$$a_4 = \sigma(0) = -18.$$

$$a_3 = \frac{\partial \sigma}{\partial s}(0) \cdot \rho_1(0) = 5 \cdot 8^{1/4}.$$

$$a_2 = \frac{1}{2!} \cdot \frac{\partial}{\partial s} \left( \frac{\partial \sigma}{\partial s}(s) \cdot \rho_1(s)^2 \right) (0) = \frac{307\sqrt{2}}{8}.$$

$$a_1 = \frac{1}{3!} \cdot \frac{\partial^2}{\partial s^2} \left( \frac{\partial \sigma}{\partial s}(s) \cdot \rho_1(s)^3 \right) (0) = -\frac{4317 \cdot 8^{3/4}}{512}.$$

$$a_0 = \frac{1}{4!} \cdot \frac{\partial^3}{\partial s^3} \left( \frac{\partial \sigma}{\partial s}(s) \cdot \rho_1(s)^4 \right) (0) = -\frac{1251}{32}.$$

- Parameterization  $\tilde{\mathcal{P}}_1(t) = \left( t^4, -18t^4 + 5 \cdot 8^{1/4}t^3 + \frac{307\sqrt{2}}{8}t^2 - \frac{4317 \cdot 8^{3/4}}{512}t - \frac{1251}{32} \right)$  is proper (see Remark 3).

**Roots  $\tau_2 = 1$ :** We have that  $\rho_2(s) = \frac{s^4 - s^3 + 5s^2 + 2s + 1}{s^4(s - 2)}$ .

- The parameterization coefficients are calculated, knowing that the multiplicity of  $\tau_2$  is  $n_2 = 1$ .

$$a_1 = \sigma(1) = -\frac{47}{8}.$$

$$a_0 = \frac{\partial \sigma}{\partial s}(1) \cdot \rho_2(1) = -\frac{363}{8}.$$

- The parameterization  $\tilde{\mathcal{P}}_2(t) = \left( t, -\frac{47}{8}t - \frac{363}{8} \right)$  is proper (see Remark 3).

**Root  $\tau_3 = 2$ :** We have that  $\rho_3(s) = \frac{s^4 - s^3 + 5s^2 + 2s + 1}{27s^4(s - 1)}$ .

- The parameterization coefficients are calculated, knowing that the multiplicity of  $\tau_3$  is  $n_3 = 1$ .

$$a_1 = \sigma(2) = -\frac{70}{33}.$$

$$a_0 = \frac{\partial \sigma}{\partial s}(2) \cdot \rho_2(2) = \frac{457}{2376}.$$

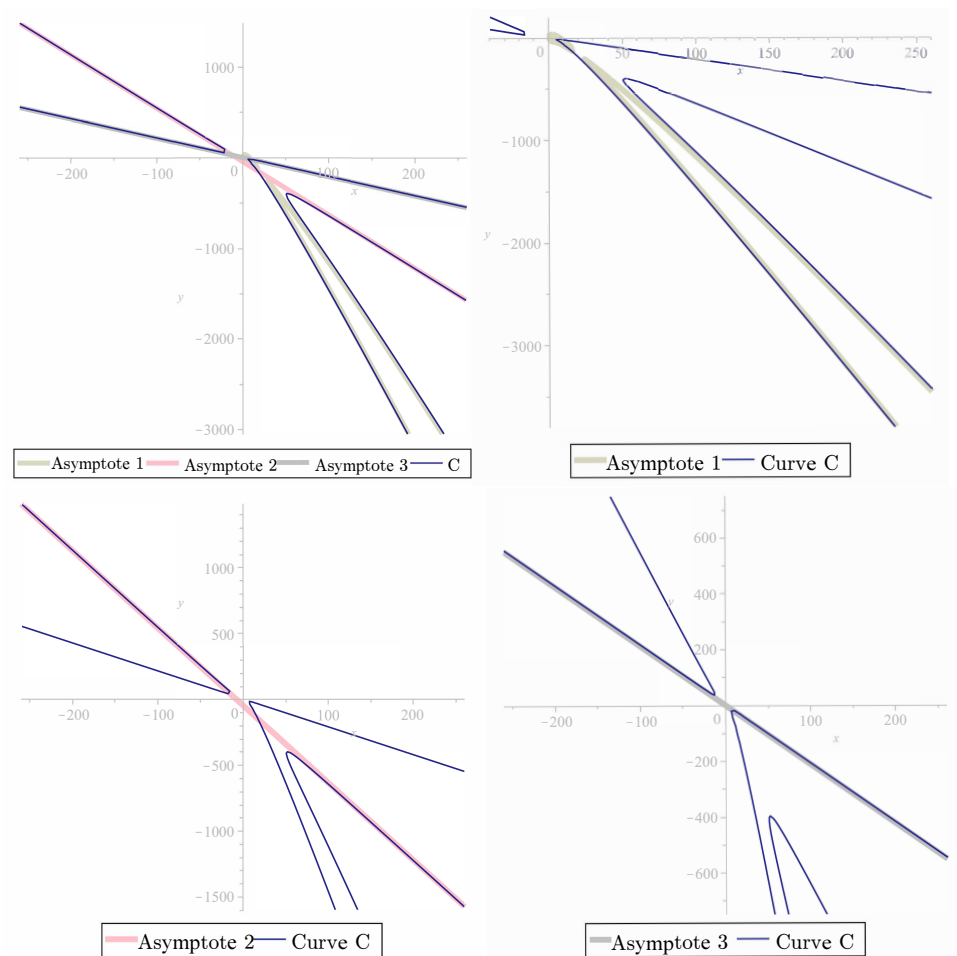
- The parameterization  $\tilde{\mathcal{P}}_3(t) = \left( t, -\frac{70}{33}t + \frac{457}{2376} \right)$  is proper (see Remark 3).

Then, the asymptotes  $\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2$  and  $\tilde{\mathcal{C}}_3$  are defined by the proper parametrizations:

$$\tilde{\mathcal{P}}_1(t) = \left( t^4, -18t^4 + 5 \cdot 8^{1/4}t^3 + \frac{307\sqrt{2}}{8}t^2 - \frac{4317 \cdot 8^{3/4}}{512}t - \frac{1251}{32} \right),$$

$$\tilde{\mathcal{P}}_2(t) = \left( t, -\frac{47}{8}t - \frac{363}{8} \right), \quad \tilde{\mathcal{P}}_3(t) = \left( t, -\frac{70}{33}t + \frac{457}{2376} \right).$$

Figure 4 plots the curve  $\mathcal{C}$  and the asymptotes  $\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2$ , and  $\tilde{\mathcal{C}}_3$ . Furthermore, it illustrates in detail the behavior of each asymptote.



**Figure 4.** Asymptotes  $\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2$  and  $\tilde{\mathcal{C}}_3$  of the curve  $\mathcal{C}$  calculated with derivatives.

To finish this section, we note that Theorem 2 can be adapted to calculate the associated coefficients to terms with a negative exponent. In this way, in addition to the asymptotes, the branches of a plane algebraic curve can be calculated. One may reason analogously with the case of limits.

#### 4. Analysis of the Computational Performance

This section discusses the performance of the previous algorithms, which construct the  $g$ -asymptotes from an irreducible plane algebraic curve. We recall that the first algorithm calculates the  $g$ -asymptotes of a curve from its implicit expression, considering the infinity branches that converge to the given curve. The other three methods deal with the case of parametrically defined curves. Note that the first two algorithms are based on the computation of the Puiseux series.

For this purpose, fourteen study cases were randomly selected with different characterizations (see Appendix B). All these curves constitute the study cases to analyze the computational performance of the algorithms presented in Section 3, which were programmed with the algebra system Maple, according to the implementation presented in Appendix C for implicit and parametric input expressions.

Under these conditions, an analysis of the computational performance of each one of the algorithms presented was carried out (based on the outlines of [28–31] to validate the performance of algorithms). The hardware overload degree of the machine resources (CPU and memory) was computed, estimating the time of use of the microprocessor and the amount of memory used for the execution of each one of the procedures defined in Appendix C. In addition, the value of real time invested by the machine was calculated for the methods presented in Section 3.

To quantify these results, it is necessary to use the tools provided by the CodeTools package of Maple. Thus, it is important to clarify that the command `CodeTools:-Usage` differentiates between the CPU time and the execution real time of a process. Thus, CPU time is the amount of time used to execute a procedure. On the other hand, the real execution time calculates the period during which the process remains in the system, from the time it is launched until it is finished, that is, the entire amount of time in which it is using the hardware resources of the system: CPU, memory, input/output, and so on.

Note that on monoprocessor systems, the total CPU usage time will always be less than the real execution time. However, on multiprocessor systems, threads could be spread across multiple cores or CPUs. In this case, the sum of all the usage times of the multiple cores, or processors, is considered as the total CPU usage time, and thus can account for the case that it is greater than the real execution time of the process.

To calculate a precise overload degree, each algorithm under analysis was iterated one hundred times for each one of the study curves. Consequently, it was possible to record reliable results on the system requirements, which are necessary to calculate the asymptotes of a given curve. The comparative criteria are based on the different properties of the input curves, such as the degree and the number of terms, as well as the following results obtained after each execution: number of infinity branches, highest degree of the analyzed asymptotes, number of real asymptotes, and number of complex asymptotes.

It is important to underline that Algorithm 1 and Algorithm 2 compute the infinity branches of the input curve by Puiseux series expansion, using the command `algcurves:-puiseux` from the algebra software Maple, with the desired accuracy fixed to ten for the calculation of the Puiseux series expansions.

From a hardware point of view, the processes were executed by a 2018 Mac Book Pro, with an Intel Core i5 processor, with four cores 2.3 GHz, 16 GiB of 2133 MHz LPDDR3 memory, Intel Iris graphics card Plus 1536 MiB Graphics 655, and 500 GiB SSD. The computer algebra software Maple 2021.1 was run on the operating system macOS Monterrey, version 12.3.1.

The following subsections present the analysis of the results obtained after the application of the methods developed for the calculation of the asymptotes of plane algebraic

curves, each of them with a different characterization. Finally, in Section 5 a comparative analysis of the algorithms is carried out.

#### 4.1. Computation of Asymptotes of Implicitly Algebraic Curves

This subsection presents the results of running Algorithm 1 one hundred times, with accuracy equal to ten for the Puiseux series expansion, on the fourteen curves presented in Appendix B. We compute the implicit polynomial of a parametric curve using resultants (see [32]).

Table 1 shows the properties for the input curves and their asymptotes. The curve with the highest degree and the highest number of terms is  $C_3$ . In this case, one may check that the asymptotes have degrees 1 and 3, much lower than the degree of  $C_3$ , which is 17.

We observe that the curve  $C_4$  is the only one with the lowest degree and the fewest number of terms. This curve has two infinity branches and the highest degree of the asymptote is 2.

Table 2 shows that the curve  $C_3$  requires the longest execution time and generates the highest overhead in the microprocessor and memory when Algorithm 1 is run. Note that the highest consumption of hardware resources corresponds to the running of the algorithm with this curve, which is the one with the highest degree and the highest number of terms. In this case, the execution consumes 283.47 ms of real time in the system and requires 362.72 ms of CPU usage and 10.55 MiB of memory.

**Table 1.** Properties of the implicit curves and their asymptotes.

Id	Degree	# Terms	# Branch	Max. Deg.	# Real Asymp	# Complex Asymp
$C_1$	7	21	5	2	3	2
$C_2$	7	21	3	3	3	-
$C_3$	17	114	5	3	1	4
$C_4$	3	8	2	2	2	-
$C_5$	6	25	3	4	3	-
$C_6$	4	11	3	2	3	-
$C_7$	4	12	2	3	2	-
$C_8$	5	13	2	3	2	-
$C_9$	5	15	2	2	2	-
$C_{10}$	7	19	3	3	1	2
$C_{11}$	15	58	4	5	2	2
$C_{12}$	13	31	7	2	1	6
$C_{13}$	9	22	5	2	3	2
$C_{14}$	5	15	2	2	2	-

**Table 2.** Hardware resources required by Algorithm 1 for the case of implicit curves.

Id	CPU Time	Real Time	Memory Used
$C_1$	26.17 ms	25.06 ms	1978.84 KiB
$C_2$	66.13 ms	31.56 ms	1386.33 KiB
$C_3$	362.72 ms	283.47 ms	10,806.75 KiB
$C_4$	14.86 ms	16.88 ms	810.67 KiB
$C_5$	168.47 ms	124.77 ms	3758.74 KiB
$C_6$	8.78 ms	8.09 ms	695.43 KiB
$C_7$	27.70 ms	39.57 ms	1495.48 KiB
$C_8$	76.90 ms	40.75 ms	1924.63 KiB
$C_9$	86.23 ms	45.01 ms	1755.44 KiB
$C_{10}$	75.12 ms	51.87 ms	1333.29 KiB
$C_{11}$	119.21 ms	82.42 ms	3706.32 KiB
$C_{12}$	8.65 ms	7.93 ms	782.95 KiB
$C_{13}$	7.13 ms	6.88 ms	602.33 KiB
$C_{14}$	76.67 ms	52.47 ms	1875.11 KiB



It is convenient to highlight the results concerning the curves  $\mathcal{C}_{12}$  and  $\mathcal{C}_{13}$  of degrees 13 and 9, respectively. Algorithm 1 presented the highest performance, and the obtained asymptotes has a degree of 2. In particular, the curve  $\mathcal{C}_{12}$  has six complex asymptotes and one real linear asymptote.

In a direct method, in order to compute the complex asymptotes, one would introduce algebraic numbers during the computations. However, for these cases, we collect the points whose coordinates depend algebraically on all the conjugate roots of a same irreducible polynomial,  $m(t) \in \mathbb{R}[t]$ . This method is based on the notion of family of conjugate points (see e.g., [32]). This concept allows us to determine the asymptotes of a curve without directly introducing algebraic numbers in the computations.

Finally, results for the curve  $\mathcal{C}_{13}$  of degree 9 show three real linear asymptotes and two complex asymptotes. Similarly, as in the previous curve, we have to collect the points whose coordinates depend algebraically on all the conjugate roots of a same irreducible polynomial  $m(t) \in \mathbb{R}[t]$ .

Thus, regarding the use of resources, Table 2 shows that curve  $\mathcal{C}_3$  requires the highest resources from the system, and it produces the most overload on the microprocessor and memory, while curves  $\mathcal{C}_{12}$  and  $\mathcal{C}_{13}$  require the least amount of system resources.

Finally, it is convenient to highlight the particularities of the curves  $\mathcal{C}_{12}$  and  $\mathcal{C}_{13}$ . Note that, in both cases, the algorithm used the least amount of time to calculate the respective asymptotes of both curves. Furthermore, the highest degree of the asymptotes constructed with Algorithm 1 is 2, much lower than the 13 and 9 degrees of the input curves  $\mathcal{C}_{12}$  and  $\mathcal{C}_{13}$ , respectively.

#### 4.2. Computation of Asymptotes of Parametric Algebraic Curves

This subsection presents the results obtained by executing the methods described in Appendix C, with an accuracy equal to ten for the Puiseux series expansion. We consider the parametric curves defined in Appendix B, given according to Equation (7):

$$\mathcal{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \quad p_i(s) = p_{i1}(s)/p_{i2}(s), \quad \gcd(p_{i1}(s), p_{i2}(s)) = 1, \quad i = 1, 2. \quad (7)$$

Table 3 shows the same results as the implicit case, except for the # Terms row.

**Table 3.** Properties of the implicit curves and their asymptotes.

Id	Degree	# Terms	# Branch	Max. Deg.	# Real Asymp	# Complex Asymp
$\mathcal{C}_1$	7	5	5	2	3	2
$\mathcal{C}_2$	7	3	3	3	3	-
$\mathcal{C}_3$	17	5	5	3	1	4
$\mathcal{C}_4$	3	3	2	2	2	-
$\mathcal{C}_5$	6	5	3	4	3	-
$\mathcal{C}_6$	4	3	3	2	3	-
$\mathcal{C}_7$	4	3	2	3	2	-
$\mathcal{C}_8$	5	3	2	3	2	-
$\mathcal{C}_9$	5	3	2	2	2	-
$\mathcal{C}_{10}$	7	3	3	3	1	2
$\mathcal{C}_{11}$	15	4	4	5	2	2
$\mathcal{C}_{12}$	13	3	7	2	1	6
$\mathcal{C}_{13}$	9	2	5	2	3	2
$\mathcal{C}_{14}$	5	2	2	2	2	-

Table 4 shows that, for the case of curves expressed in parametric form, Algorithm 2 presents a lower efficiency than Algorithm 1. In most cases, this algorithm produces a higher system overhead and requires more hardware resources. Thus, it can be seen that Algorithm 2 generates the highest real execution times, requiring a greater time of use of the microprocessor and a bigger memory capacity for all the study cases.

**Table 4.** Hardware resources used by Algorithm 2 for the case of parametric curves.

Id	CPU Time	Real Time	Memory Used
$C_1$	16.13 ms	20.38 ms	2014.46 KiB
$C_2$	131.45 ms	117.78 ms	17,305.41 KiB
$C_3$	$>10^8$ ms	$>10^8$ ms	$>10^8$ KiB
$C_4$	3.98 ms	3.60 ms	276.03 KiB
$C_5$	114.12 ms	63.74 ms	10,630.93 KiB
$C_6$	2.57 ms	2.69 ms	288.92 KiB
$C_7$	25.10 ms	26.58 ms	3193.96 KiB
$C_8$	102.26 ms	72.53 ms	8634.75 KiB
$C_9$	47.83 ms	25.44 ms	2832.28 KiB
$C_{10}$	18,496.00 ms	16,028.00 ms	1,795,206.41 KiB
$C_{11}$	$>10^8$ ms	$>10^8$ ms	$>10^8$ KiB
$C_{12}$	21,726.00 ms	18,961.00 ms	2,720,673.58 KiB
$C_{13}$	22,877.00 ms	17,054.00 ms	2,667,242.91 KiB
$C_{14}$	24.00 ms	24.00 ms	3668.91 KiB

The greatest hardware resource requirements correspond to the calculation of the asymptotes of the curves  $C_3$  and  $C_{11}$ , where the algorithm needed more than 24 h of CPU usage and more than 100 GiB of memory.

Moreover, the execution of the algorithm with the input curve  $C_{13}$  requires 17 s of real time, 23 s of microprocessor usage time, and 2.55 GiB of memory; which contrasts sharply with applying Algorithm 1 to the implicit curve  $C_{13}$ , with a real execution time of 6.88 ms, 7.13 ms of CPU usage and 602.33 KiB of memory.

On the other hand, it is interesting to note that the application of Algorithm 2 for the calculation of the asymptotes of the curve  $C_4$  presents the least degree of overload, even compared with the application of Algorithm 1 to the case of the implicit curve  $C_4$ .

However, if these results are compared with those of Table 2, it is observed that as the complexity of the curve increases, the CPU usage time, the real execution time, and the memory needs increase exponentially.

#### 4.3. Computation of Asymptotes of Parametric Algebraic Curves with Limits

Table 5 shows the results of executing Algorithm 3 to construct the asymptotes of the curves in Appendix B, from the parametric expressions. Note that this result is the same as the one in Table 3.

**Table 5.** Properties of parametric curves and their asymptotes.

Id	Input Curve			Output Asymptotes		
	Deg.	N.Terms	N.Branches	Max. Deg.	N.Real Asymp	N.Complex Asymp
$C_1$	7	5	5	2	3	2
$C_2$	7	3	3	3	3	0
$C_3$	17	5	5	3	1	4
$C_4$	3	3	2	2	2	0
$C_5$	6	5	3	4	3	0
$C_6$	4	3	3	2	3	0
$C_7$	4	3	2	3	2	0
$C_8$	5	3	2	3	2	0
$C_9$	5	3	2	1	2	0
$C_{10}$	7	3	3	3	1	2
$C_{11}$	15	4	4	5	2	2
$C_{12}$	13	3	7	2	1	6
$C_{13}$	9	2	5	2	3	2
$C_{14}$	5	2	2	2	2	0

Table 6 allows us to review the degree of overload that this algorithm produces in the system.

**Table 6.** Hardware resources required by Algorithm 3 for parametric curves.

Id	t. CPU	t. Stay Actual	Memory Used
$C_1$	41.74 ms	44.93 ms	4815.36 KiB
$C_2$	11.99 ms	13.44 ms	1241.23 KiB
$C_3$	59.25 ms	62.82 ms	5910.91 KiB
$C_4$	7.77 ms	11.40 ms	587.45 KiB
$C_5$	23.00 ms	18.78 ms	1836.80 KiB
$C_6$	20.78 ms	11.31 ms	832.64 KiB
$C_7$	9.76 ms	10.73 ms	871.49 KiB
$C_8$	11.10 ms	11.56 ms	973.96 KiB
$C_9$	5.92 ms	6.48 ms	527.17 KiB
$C_{10}$	15.30 ms	16.50 ms	1302.70 KiB
$C_{11}$	42.77 ms	31.43 ms	2796.50 KiB
$C_{12}$	48.13 ms	34.53 ms	3242.84 KiB
$C_{13}$	22.06 ms	17.93 ms	1390.01 KiB
$C_{14}$	8.32 ms	8.49 ms	636.6 KiB

Table 6 shows a greater general efficiency in managing the computer's memory than Algorithms 1 and 2. Improvements are also observed in the times of CPU usage and real stay in the system, highlighting the behavior of the curves  $C_3$  and  $C_9$ .

The first case, the curve  $C_3$ , corresponds to the curve for which Algorithm 3 required the greatest real time and generated the greatest overload in the microprocessor with the greatest need for memory. However, in this case, the resource usage is much lower than the previous results of Tables 2 and 4, with only 62.82 ms residence time in the system, 59.25 ms of CPU usage, and 1.21 MiB of memory.

In the second case, the curve  $C_9$ , the lowest overhead was obtained with only 6.48 ms of real time in the system, 5.92 ms of CPU usage, and 0.5 MiB of memory. Thus, this curve, of degree five, shows the least degree of overload when constructing its two real asymptotes with Algorithm 3.

#### 4.4. Computation of Asymptotes of Parametric Algebraic Curves with Derivatives

Table 7 shows the results of executing Algorithm 4 to construct the curve asymptotes of Appendix B, using some derivatives of univariate functions. Note that this result is consistent with that of Tables 3 and 5.

**Table 7.** Properties of parametric curves and their asymptotes.

Id	Input Curve			Output Asymptotes		
	Deg.	N.Terms	N.Branches	Max. Deg.	N.Real Asymp	N.Complex Asymp
$C_1$	7	5	5	2	3	2
$C_2$	7	3	3	3	3	0
$C_3$	17	5	5	3	1	4
$C_4$	3	3	2	2	2	0
$C_5$	6	5	3	4	3	0
$C_6$	4	3	3	2	3	0
$C_7$	4	3	2	3	2	0
$C_8$	5	3	2	3	2	0
$C_9$	5	3	2	1	2	0
$C_{10}$	7	3	3	3	1	2
$C_{11}$	15	4	4	5	2	2
$C_{12}$	13	3	7	2	1	6
$C_{13}$	9	2	5	2	3	2
$C_{14}$	5	2	2	2	2	0

Table 8 shows the degree of overload that this algorithm produces in the system. In this case, a very significant improvement is presented with respect to the previous cases, Algorithms 1–3.

**Table 8.** Hardware resources required by Algorithm 4 for parametric curves.

Id	t. CPU	t. Stay Actual	Memory Used
$C_1$	10.18 ms	11.13 ms	803.07 KiB
$C_2$	4.16 ms	5.61 ms	347.86 KiB
$C_3$	40.48 ms	35.30 ms	5155.88 KiB
$C_4$	2.55 ms	2.68 ms	223.10 KiB
$C_5$	4.17 ms	5.11 ms	331.83 KiB
$C_6$	3.42 ms	3.38 ms	284.55 KiB
$C_7$	3.37 ms	3.42 ms	266.60 KiB
$C_8$	3.31 ms	3.26 ms	268.50 KiB
$C_9$	3.07 ms	3.18 ms	263.64 KiB
$C_{10}$	5.38 ms	6.58 ms	421.00 KiB
$C_{11}$	50.00 ms	48.29 ms	7446.49 KiB
$C_{12}$	13.13 ms	14.51 ms	1234.98 KiB
$C_{13}$	8.83 ms	9.14 ms	715.44 KiB
$C_{14}$	3.26 ms	3.23 ms	265.36 KiB

In almost all the input curves, except for the case of  $C_{11}$ , the use of hardware resources, time of use of the microprocessor, time of real stay in the system, and memory capacity, show the highest efficiency.

The exception occurs for the case of the construction of the asymptotes of the curve  $C_{11}$ , which requires more memory, more CPU time, and stays longer in the system than for the case of Algorithm 3.

Note that the best performance is obtained by constructing the curve asymptotes of the simplest curve,  $C_4$ , with only 2.55 ms processor usage, 2.68 ms of permanence in the system, and 223.10 KiB of memory capacity. The curve that generates the greatest overload is  $C_{11}$ , which requires 50 ms of CPU, with a real time of 48.29 ms and 7.27 MiB of memory.

We briefly analyze the particular case of the asymptotes obtained for the curve  $C_{11}$ . This curve has a complex asymptote of degree five and two real asymptotes of degrees three and two. Similarly as above, in this case, we need to introduce conjugate points, that is, we collect the points whose coordinates depend algebraically on all the conjugate roots of a same irreducible polynomial,  $m(t) \in \mathbb{R}[t]$ . Note that working with the conjugate points of the complex asymptote introduces the greatest degree of overhead.

## 5. Results and Discussion

In this section, a comparative analysis of Algorithms 1–4 is presented. For this purpose, some properties of the input curves are considered, as well as the results obtained when constructing the respective asymptotes (see Appendix B).

Table 9 allows to review the efficiency of each algorithm and establishes comparative criteria regarding the time of use of the CPU from the simplest curve that is, the one with the lowest degree and the fewest number of terms,  $C_4$ , with respect to the curve of greater complexity,  $C_3$ .

Table 9 compares the CPU usage times after the execution of each of the four algorithms used for the construction of asymptotes of the curves  $C_1$ – $C_{14}$ . The worst results are highlighted in red, and the five best CPU usage times are highlighted in green, given mainly when Algorithm 4 constructs the asymptotes of the curves  $C_4$ ,  $C_8$ ,  $C_9$ , and  $C_{14}$ .

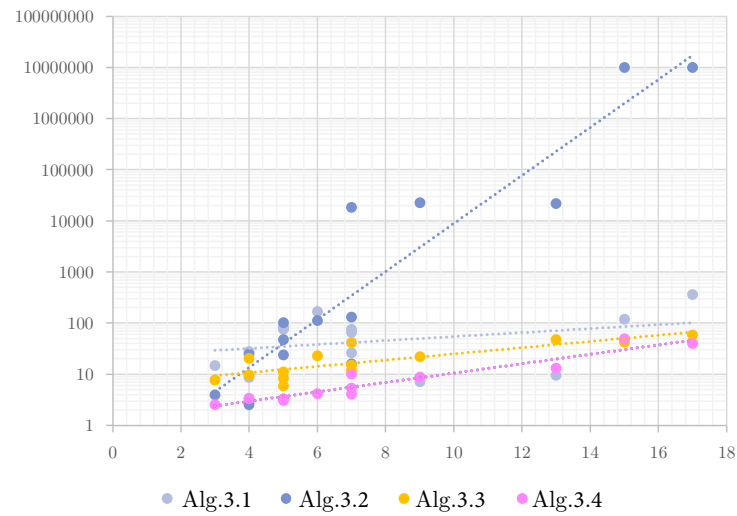
Figure 5 represents the behavior of each of the algorithms, depending on the degree of the input curve (axis  $x$ ), with respect to the CPU usage time (axis  $y$ ) expressed in milliseconds. Note that this is a linear–logarithmic plot, based on 10, in which the trend lines are exponential.

Analyzing Algorithm 4, the highest efficiency is observed, since it requires the least amount of CPU usage time. Likewise, it is also shown that, for all algorithms, the degree of the curve determines the microprocessor time needed to build the asymptotes.

Table 9. Properties of the input curve and microprocessor time needed to execute Algorithms 1–4.

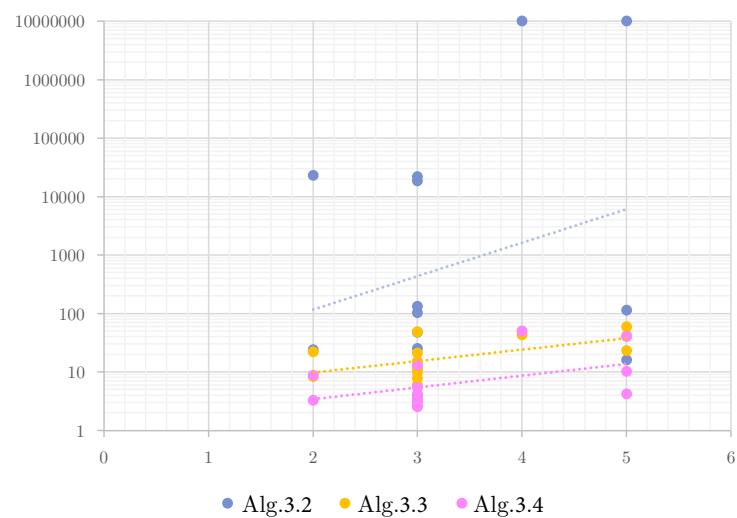
Id	Deg.	Number of Terms			Asymptotes			CPU Time (ms)			
		Implicit	Parametric	Deg. Max.	N.Asymp	N.Real Asymp	N.Complex Asymp	Alg.3.1	Alg.3.2	Alg.3.3	Alg.3.4
$C_1$	7	21	5	2	5	3	2	26.17	16.13	41.74	10.18
$C_2$	7	21	3	3	3	3	0	66.13	131.45	11.99	4.16
$C_3$	17	114	5	3	5	1	4	362.72	>10 <sup>8</sup>	59.25	40.48
$C_4$	3	8	3	2	2	2	0	14.86	3.98	7.77	<b>2.55</b>
$C_5$	6	25	5	4	3	3	0	168.47	114.12	23.00	4.17
$C_6$	4	11	3	2	3	3	0	8.78	<b>2.57</b>	20.78	3.42
$C_7$	4	12	3	3	2	2	0	27.70	25.10	9.76	3.37
$C_8$	5	13	3	3	2	2	0	76.90	102.26	11.10	<b>3.31</b>
$C_9$	5	15	3	1	2	2	0	86.23	47.83	5.92	<b>3.07</b>
$C_{10}$	7	19	3	3	3	1	2	75.12	18,496.00	15.30	5.38
$C_{11}$	15	58	4	5	4	2	2	119.21	>10 <sup>8</sup>	42.77	50.00
$C_{12}$	13	31	3	2	7	1	6	8.65	21,726.00	48.13	13.13
$C_{13}$	9	22	2	2	5	3	2	7.13	22,877.00	22.06	8.83
$C_{14}$	5	15	2	2	2	2	0	76.67	24.00	8.32	<b>3.26</b>

On the other hand, it is noteworthy that although Algorithm 2 behaves well with simple curves (see curve  $C_6$  in Table 9), CPU usage time increases exponentially as the degree of the input curve increases (see curves  $C_3$  and  $C_{11}$  in Table 9).



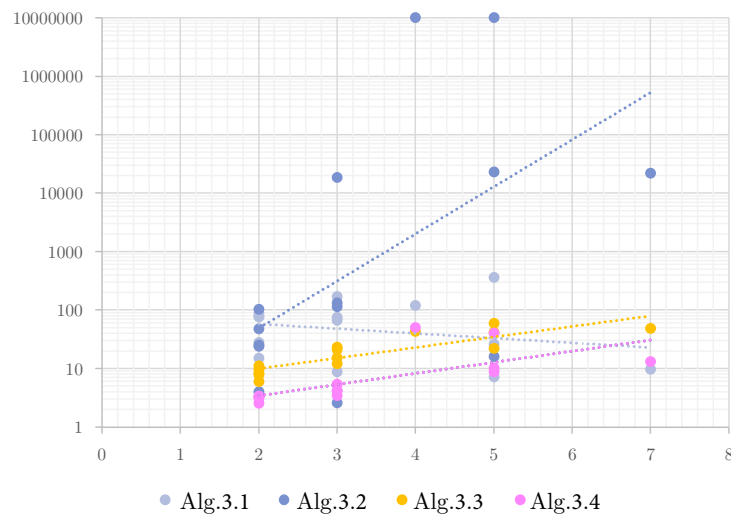
**Figure 5.** Microprocessor usage time (ms) depending on the degree.

Figure 6 shows how as the number of terms of the input curve increases the CPU time needed to execute Algorithm 3. However, the density of the input curve seems to affect the efficiency of the rest of the algorithms analyzed to a lesser extent. Note that, in this case, Algorithm 1 is not evaluated. This is due to the fact that the implicit case does not allow establishing a comparative criterion with the rest of the algorithms in what refers to the density of the input curve.



**Figure 6.** Microprocessor usage time (ms) depending on the number of terms.

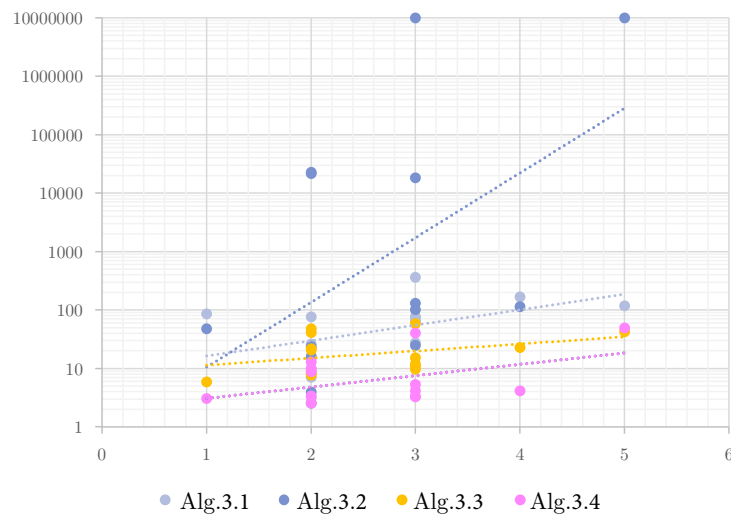
On the other hand, the same thing happens in Figure 7. Algorithm 2 requires more CPU time in cases where a high number of branches is observed (remember that trend lines are exponential). However, it does not seem that the number of branches determines the time of use of the microprocessor in the case of the other algorithms represented.



**Figure 7.** Microprocessor usage time (ms) depending on the number of branches.

Figure 8 shows the CPU time required by each algorithm, as a function of the highest degree of the asymptotes, of each of the input curves. It can be seen that Algorithm 4 is the one that, in general, presents the highest efficiency, since it requires the least amount of CPU time.

Likewise, it is also shown that Algorithm 2 has a quite disparate behavior, since it does not seem that the degree of the asymptotes affects the CPU usage time for this algorithm.



**Figure 8.** Microprocessor usage time depending on the highest degree of asymptotes.

Table 10 shows the real time it takes to execute each algorithm for each of the curves, with results very similar to those obtained in Table 9, which is quite logical because both times are related.

We recall that real time refers to the period from when the process starts until it ends, that is, the time it remains in the system. Note that, in some cases, this time is less than the microprocessor usage time, because the Maple computer algebra software executes on a multicore system. In this case, the CPU usage time is the sum of all the usage times of the different microprocessor cores. This sum is what determines the total CPU value given in Table 9.

In Table 10, the worst global results are highlighted in red, and they correspond to Algorithm 2, executed with the input curves  $C_3$  and  $C_{11}$ . In addition, the five best CPU usage times are shown in green, and they correspond to Algorithm 2, executed with the input curve  $C_6$  and when the asymptotes of the curves  $C_4$ ,  $C_8$ ,  $C_9$ , and  $C_{14}$  are determined with Algorithm 4.



It can be verified that these results show an efficiency similar to that obtained when CPU usage time is computed (see Table 9).

Finally, Table 11 allows comparing the memory needs of the analyzed algorithms. The worst results are highlighted in red, and the five best CPU usage times are highlighted in green. These times correspond to Algorithm 4 when constructing the asymptotes of the curves  $C_4$ ,  $C_7$ ,  $C_8$ ,  $C_9$ , and  $C_{14}$ .

With this hardware resource, the algorithm that marks the best results is Algorithm 4, followed by Algorithms 1 and 3. However, for the case of Algorithm 2, the hardware memory requirements grow exponentially with the degree of the input curve and the number of terms of the input curve.

Figures 9 and 10 reflect the behavior of each of the algorithms considering the parameters of the input curve regarding memory.

For the analysis of these figures, we remark that the  $y$ -axis is logarithmic, base 10, and that the trend lines are exponential.

On the other hand, looking at Figure 11, it does not seem that the number of branches is a determining condition for memory needs in the case of the analyzed algorithms. The same happens in Figure 12. In both cases, it can be seen that Algorithm 4 is the one that, in general, presents the highest efficiency with the least amount of memory needed to be able to execute it. However, Algorithm 2 has quite disparate behavior, since neither the number of branches nor the highest degree of the asymptotes seems to determine the memory requirements to execute the algorithm. For example, it can be seen that for asymptotes whose maximum degree is two, the memory needs can range between 276.03 KiB for the curve  $C_4$  and 2.59 GiB for the curve  $C_{12}$ .

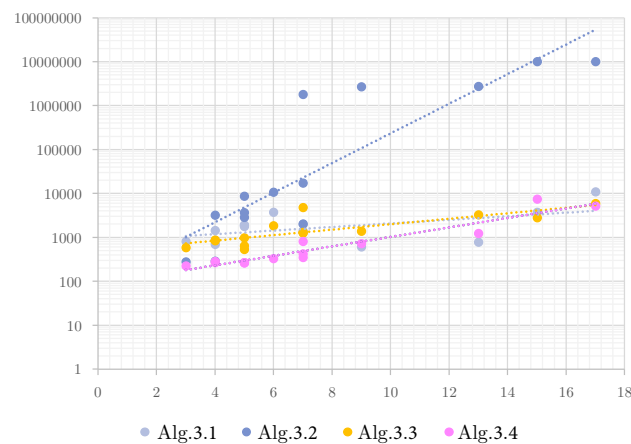


Figure 9. Required memory capacity (KiB) depending on degree.

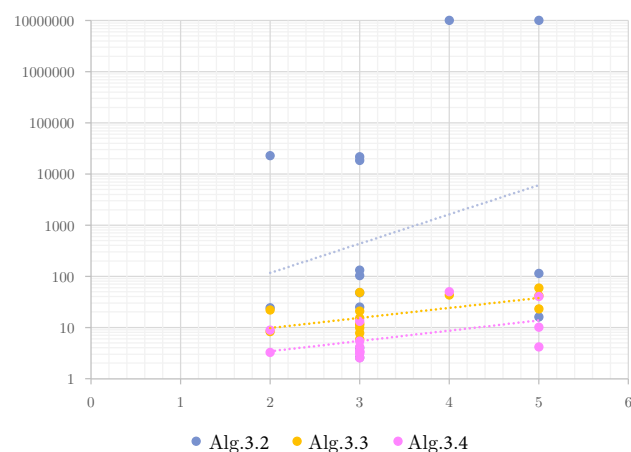


Figure 10. Required memory capacity (KiB) depending on the number of terms.

Table 10. Properties of the input curve and real time of stay in the system necessary to execute Algorithms 1–4.

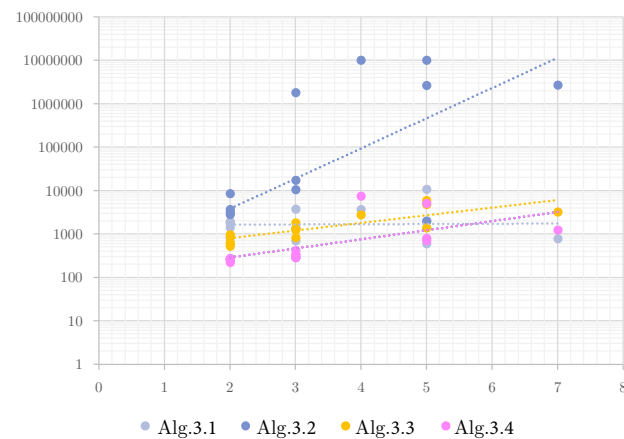
Id	Deg.	Number of Terms			Asymptotes			Real Time Spent in the System (ms)			
		Implicit	Parametric	Max. Deg.	N.Asymp	N.Real Asymp	N.Complex Asymp	Alg.3.1	Alg.3.2	Alg.3.3	Alg.3.4
$C_1$	7	21	5	2	5	3	2	25.06	20.38	44.93	11.13
$C_2$	7	21	3	3	3	3	0	31.56	117.78	13.44	5.61
$C_3$	17	114	5	3	5	1	4	283.47	>10 <sup>8</sup>	62.82	35.3
$C_4$	3	8	3	2	2	2	0	16.88	3.6	11.4	<b>2.68</b>
$C_5$	6	25	5	4	3	3	0	124.77	63.74	18.78	5.11
$C_6$	4	11	3	2	3	3	0	8.09	<b>2.69</b>	11.31	3.38
$C_7$	4	12	3	3	2	2	0	39.57	26.58	10.73	3.42
$C_8$	5	13	3	3	2	2	0	40.75	72.53	11.56	<b>3.26</b>
$C_9$	5	15	3	1	2	2	0	45.01	25.44	6.48	<b>3.18</b>
$C_{10}$	7	19	3	3	3	1	2	51.87	16,028	16.5	6.58
$C_{11}$	15	58	4	5	4	2	2	82.42	>10 <sup>8</sup>	31.43	48.29
$C_{12}$	13	31	3	2	7	1	6	7.93	18,961	34.53	14.51
$C_{13}$	9	22	2	2	5	3	2	6.88	17,054	17.93	9.14
$C_{14}$	5	15	2	2	2	2	0	52.47	24.00	8.49	<b>3.23</b>

Table 11. Properties of the input curve and memory capacity needed to execute Algorithms 1–4.

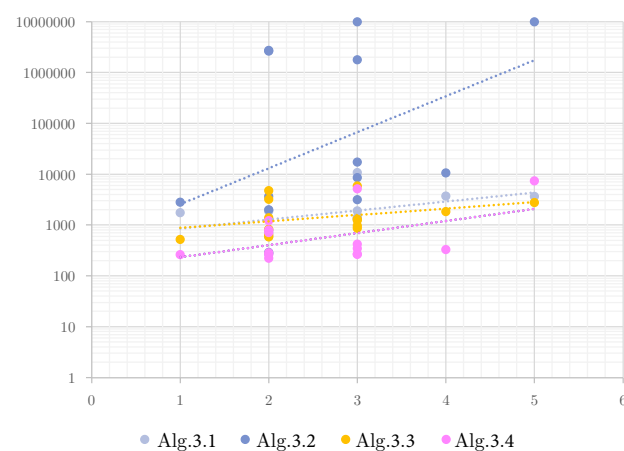
Id	Deg.	N. of Terms			Asymptotes			Memory Capacity (KiB)			
		Impl.	Param.	Max. Deg.	N.Asymp	N.Real Asymp	N.Complex Asymp	Alg.3.1	Alg.3.2	Alg.3.3	Alg.3.4
$C_1$	7	21	5	2	5	3	2	1978.84	2014.46	4815.36	803.07
$C_2$	7	21	3	3	3	3	0	1386.33	17,305.41	1241.23	347.86
$C_3$	17	114	5	3	5	1	4	10,806.75	>10 <sup>8</sup>	5910.91	5155.88
$C_4$	3	8	3	2	2	2	0	810.67	276.03	587.45	<b>223.10</b>
$C_5$	6	25	5	4	3	3	0	3758.74	10,630.93	1836.80	331.83
$C_6$	4	11	3	2	3	3	0	695.43	288.92	832.64	284.55
$C_7$	4	12	3	3	2	2	0	1425.48	3193.96	871.49	<b>266.60</b>
$C_8$	5	13	3	3	2	2	0	1924.63	8634.75	973.96	<b>268.50</b>
$C_9$	5	15	3	1	2	2	0	1755.44	2832.28	527.17	<b>263.64</b>
$C_{10}$	7	19	3	3	3	1	2	1333.29	1,795,206.41	1302.70	421.00
$C_{11}$	15	58	4	5	4	2	2	3706.32	>10 <sup>8</sup>	2796.50	7446.49
$C_{12}$	13	31	3	2	7	1	6	782.95	2,720,673.58	3242.84	1234.98
$C_{13}$	9	22	2	2	5	3	2	602.33	2,667,242.91	1390.01	715.44
$C_{14}$	5	15	2	2	2	2	0	1875.11	3668.91	636.60	<b>265.36</b>

Finally, it can be stated that Algorithm 4 is the one that presents the best computational performance, requiring the least amount of hardware resources: CPU time, real execution time, and amount of memory. Likewise, it is also shown that, for all algorithms, the degree of the curve is the parameter that determines, to a greater extent, the amount of system resources needed to build the asymptotes.

On the other hand, it is noteworthy that Algorithm 2 has a good behavior with “simple” curves’ see the values for the case of the curves  $C_4$  and  $C_6$  in Tables 9–11. However, it is a very “heavy” algorithm when the degree of the input curve increases and the number of terms increases, in which case the hardware needs of said algorithm increase exponentially (see curves  $C_3$  and  $C_{11}$  in the tables above).



**Figure 11.** Required memory capacity (KiB) depending on the number of branches.



**Figure 12.** Required memory capacity, in KiB, according to the maximum degree asymptote.

The results of this evaluation show how the Algorithms 3 and 4, additionally to improving the degree of system overload, are more efficient than Algorithms 1 and 2, mainly highlighting the results of Algorithm 4.

On the other side, Algorithm 2 is the one that requires the most resources, generating the greatest overhead in the system, mainly due to the complexity introduced by the computation of the Puiseux series. In addition, the higher the precision for obtaining the series, the greater memory capacity and longer execution time the process requires.

After carrying out this analysis, we have the following conclusions regarding the main aspects that determine the greatest overload in the system, when the g-asymptotes of a given curve are constructed with: the degree of the input curve, the number of terms of the input curve, and the degree of complex asymptotes.

On the other hand, it was observed that the precision applied when performing the Puiseux series expansion determines the memory capacity needed by the process. At higher precision, the memory requirement grows exponentially and, also, although to a lesser extent, the CPU time and the real execution time.

These results support the idea that the proposed methods can be used effectively for solving large-scale problems, without the high computational resource utilization requirements imposed by previous methods based on the development of Puiseux series. In particular, Algorithms 3 and 4 improve the efficiency of the previous methods (Algorithms 1 and 2), showing higher computational performance, requiring fewer hardware resources, and considerably reducing the degree of computer overload.

## 6. Conclusions and Future Works

The  $g$ -asymptotes determine the behavior of a curve at infinity. The importance of these entities is clear for, for instance, plotting the graph of the input curve, studying the topology, or giving some ideas to deal with some numerical problems. In addition, using these mathematical objects, predictive or trend calculation models can be established, which could improve, for instance, results of simple linear regression.

As future work, it is proposed to generalize these results to curves defined by a nonrational parametrization (see [25]), as well as to surfaces, and to study the families of existing asymptotes.

Furthermore, we think that as future works, these methods could satisfactorily be extended to enhanced fuzzy-based algorithms, interval environment or shortest path problem solving (see for instance [33–35]).

Although this article computes “real” performance and the system overload of the algorithms proposed, the point we wish to make here is the importance of time complexity, which is difficult to compute exactly. Nevertheless, we estimated some preliminary results that confirm the analytical discussion presented in this article, where the time complexity to calculate the Puiseux series is too high (see [36]) compared with the calculation of limits or derivatives (see [37]), even if they are of different orders.

Our main objective is to propose methods that can be used effectively to solve large-scale problems, including real-world cases and symmetry issues, without the high computational requirements.

**Author Contributions:** Conceptualization, S.P.-D. and M.F.d.S.; methodology, E.C.-M.; software, E.C.-M.; validation, J.R.M.B. and S.P.-D.; formal analysis, M.F.d.S.; investigation, S.P.-D.; resources, J.R.M.B.; data curation, E.C.-M.; writing—original draft preparation, E.C.-M.; writing—review and editing, S.P.-D.; visualization, J.R.M.B.; supervision, S.P.-D.; project administration, M.F.d.S.; funding acquisition, S.P.-D. All authors have read and agreed to the published version of the manuscript.

**Funding:** The author S. Pérez-Díaz is partially supported by Ministerio de Ciencia, Innovación y Universidades—Agencia Estatal de Investigación/PID2020-113192GB-I00 (Mathematical Visualization: Foundations, Algorithms and Applications). The author R. Magdalena Benedicto is partially supported by the State Plan for Scientific and Technical Research and Innovation of the Spanish MCI (PID2021-127946OB-I00).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The author S. Pérez-Díaz is partially supported by Ministerio de Ciencia, Innovación y Universidades—Agencia Estatal de Investigación/PID2020-113192GB-I00 (Mathematical Visualization: Foundations, Algorithms and Applications). The author R. Magdalena Benedicto is partially supported by the State Plan for Scientific and Technical Research and Innovation of the Spanish MCI (PID2021-127946OB-I00). The author S. Pérez-Díaz belongs to the Research Group ASYNACS (Ref. CCEE2011/R34).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

In the following Appendices, we present the parametric curves used in the study of the algorithms presented in the previous sections and the pseudocode for the algebra system Maple of the algorithms presented in this paper to construct the asymptotes of plane algebraic curves.

## Appendix B. Definition of Parametric Curves

- $\mathcal{P}_1(s) := \left( \frac{s^2 + 1}{(s^3 - s + 1)(s^2 - 1)^2}, \frac{2s^3 + 5s^2 + 1}{s^5 - 2s^3 + s^2 + s - 1} \right)$ .
- $\mathcal{P}_2(s) := \left( \frac{(10 \cdot s^3 - 1) \cdot s^2}{(s^2 - 1)^2 \cdot (s - 2)^3}, \frac{s^3 + s^2 + 3}{(s - 2)^2 \cdot (s^2 - 1)} \right)$ .
- $\mathcal{P}_3(s) := \left( \frac{s^8 + 2s^4 - s^2 - s - 1}{(s^2 + 1)^4 \cdot (s^2 + 2)^3 \cdot s^3}, \frac{s^7 + s^6 - s^5 + s^2 + 1}{(s^2 + 2)^3 \cdot s} \right)$ .
- $\mathcal{P}_4(s) := \left( \frac{5 + s^2}{s \cdot (s - 2)^2}, \frac{s^2 + 3s + 1}{s \cdot (s - 2)} \right)$ .
- $\mathcal{P}_5(s) := \left( \frac{s^4 - s^3 + 5s^2 + 2s + 1}{s^4 \cdot (s - 1) \cdot (s - 2)}, \frac{2s^4 - 3s^3 - 2s^2 - 26s - 18}{s^4 \cdot (s - 1) \cdot (s - 2)} \right)$ .
- $\mathcal{P}_6(s) := \left( \frac{s^3 + 2s - 1}{(s^2 - 1) \cdot (s - 2)^2}, \frac{2s^3 + s^2 + 1}{(s - 2) \cdot (s^2 - 1)} \right)$ .
- $\mathcal{P}_7(s) := \left( \frac{s^3 + 2s - 1}{(s - 1) \cdot (s - 2)^3}, \frac{2s^3 + s^2 + 1}{(s - 2)^2 \cdot (s - 1)} \right)$ .
- $\mathcal{P}_8(s) := \left( \frac{4 \cdot (s^2 + 1)}{s^2 \cdot (s - 2)^3}, \frac{(2 \cdot s^2 + 2 \cdot s - 1)}{s^2 \cdot (s - 2)^2} \right)$ .
- $\mathcal{P}_9(s) := \left( \frac{2 \cdot s^2 - 9}{s^2 \cdot (s + 3)^3}, \frac{s^2 + s - 1}{s^2} \right)$ .
- $\mathcal{P}_{10}(s) := \left( \frac{s^3 + 2s - 1}{(s^2 + 1)^2 \cdot (s - 2)^3}, \frac{2s^3 + s^2 + 1}{(s - 2)^2 \cdot (s^2 + 1)} \right)$ .
- $\mathcal{P}_{11}(s) := \left( \frac{s^5 + 2s^2 + s - 1}{s^2 \cdot (s - 1)^3 \cdot (s^2 + 1)^5}, \frac{2s^4 - s^3 + s^2 + 1}{(s^2 + 1)^3 \cdot (s - 1)^2} \right)$ .
- $\mathcal{P}_{12}(s) := \left( \frac{s - 1}{s \cdot (s^6 + 2)^2}, \frac{s^3 - s + 1}{(s^6 + 2)} \right)$ .
- $\mathcal{P}_{13}(s) := \left( \frac{s - 1}{(s + 1) \cdot (s^4 - 2)^2}, \frac{s^3}{s^4 - 2} \right)$ .
- $\mathcal{P}_{14}(s) := \left( \frac{s^2 + 10}{s^2 \cdot (s - 2)^3}, \frac{s^2 + 1}{s^2} \right)$ .

## Appendix C. Implementation

The following page shows the procedures which were implemented for Algorithms 1–4 with the Maple algebra software.

## Procedure AsymptotesImplicit (implicit\_curve, accuracy)

### Description

Computes the parametrizations of the g-asymptotes of an implicit curve, considering Puiseux expansion up to a fixed accuracy.

@param {polynomial} implicit\_curve  
 @param {number} accuracy  
 @return [{expression}, {expression}]... parametric\_asymptotes list

### Code

```

> AsymptotesImplicit := proc(implicit_curve, accuracy)
  local F, g;
  local x, y, nroots, i := 1;
  local points_inf_list;
  local Phi, parametric_asymptoteslist := NULL;
  local phi, l_YZ, l_aux, e_index, r, tilde_aux, tilde_r;

  x := indets(implicit_curve)[1];
  y := indets(implicit_curve)[2];
  F := numer(subs({x = X/Z, y = Y/Z}, implicit_curve));      # Projective curve
  g := subs(x=1, F);                                          # g(y,z)
  Phi := convert(puiseux(g, z=0, y, accuracy), list);
  nroots := nops(Phi);

  for i from 1 to nroots do                                  # Leaves and braches
    phi := Phi[i];
    l_YZ := phi/z;
    l_aux := subs(z = 1/z, l_YZ);
    e_index := EIndex(l_aux);
    r := simplify(subs(z = z^e_index, l_aux), radical, symbolic);
    tilde_aux := NonNegativeOps(expand(simplify(r), radical, symbolic));
    tilde_r := subs(z=t, tilde_aux);
    parametric_asymptoteslist := [t^e_index, tilde_r], parametric_asymptoteslist;
  end do;

  return [parametric_asymptoteslist];
end proc;

```

### Example

```

> f := -176 x·y3 - 88 y4 + 2352 x2·y + 1368 x·y2 + 360 y3 - 1568 x2 - 1960 x·y - 520 y2 + 896 x
  + 368 y - 128;
AsymptotesImplicit(f, 3);

```

$$\left[ [t, -2t + 3], \left[ t^2, \frac{7\sqrt{33}t}{11} + \frac{7}{33} \right], \left[ t, \frac{2}{3} \right] \right]$$

## Procedure AsymptotesParametric (parametric\_curve, accPuisseux)

### Description

Computes the parametrizations of the g-asymptotes from a parametric curve, considering Puiseux expansions up to a fixed accuracy.

@param [{expression}, {expression}] parametric\_curve  
 @param {number} accPuisseux  
 @return [{expression}, {expression}], ...] parametric\_asymptotes list

### Code

```

> AsymptotesParametric := proc(parametric_curve, accPuisseux)
  local PP, g;
  local p1, p2, ec_aux;
  local l, aux_l, aux_r, r, tilde_r;
  local s, e_index, i := 1;
  local parametric_asymptoteslist := NULL;

  s := indets(parametric_curve)[1];
  p1 := parametric_curve[1];
  p2 := parametric_curve[2];
  PP := [1,  $\frac{p2}{p1}$ ,  $\frac{1}{p1}$ ]; # Projective curve
  g := [ $\frac{p2}{p1}$ ,  $\frac{1}{p1} - t$ ];
  ec_aux := numer(g[2]);
  l := convert(puisseux(ec_aux, t=0, s, accPuisseux), set);
  for i to nops(l) do
    aux_r := subs(s=l[i], p2);
    aux_r := convert(series(aux_r, t=0, accPuisseux), polynomial);
    r := subs(t=z-1, aux_r);
    e_index := EIndex(r);
    r := expand(simplify(subs(z=ze_index, r), radical, symbolic));
    tilde_r := NonNegativeOps(r);
    parametric_asymptoteslist := [te_index, subs(z=t, tilde_r)],
    parametric_asymptoteslist;
  end do;

  return [parametric_asymptoteslist];
end proc:

```

### Example

```

> P := [ $\frac{s^3 + 2s - 1}{(s^2 - 1) \cdot (s - 2)^2}$ ,  $\frac{2s^3 + s^2 + 1}{(s - 2) \cdot (s^2 - 1)}$ ]:
parametric_asymptotesL := AsymptotesParametric(P, 4);

```

$$parametric\_asymptotesL := \left[ \left[ t^2, \frac{7\sqrt{3}\sqrt{11}t}{11} + \frac{7}{33} \right], [t, -2t + 3], \left[ t, \frac{2}{3} \right] \right]$$



## Procedure AsymptotesParametricLimits (parametric\_curve)

### Description

Computes the parametrization of the g-asymptotes of a given parametric curve using limits.

@param [{expression}, {expression}] parametric\_curve  
@return [{expression}, {expression}], ... parametric\_asymptoteslist

### Code

```

> AsymptotesParametricLimits := proc(parametric_curve)
  local s, τ, ρ, n1, n2, i := 1, j, aux;
  local p1, p2, tilde_r;
  local parametric_asymptote;
  local τlist, ρlist, parametric_asymptoteslist := NULL;
  local f, a;

  s := indets(parametric_curve)[1];
  p1 := parametric_curve[1];
  p2 := parametric_curve[2];
  τlist := RootsMultiplicityParametric(parametric_curve, 1);
  ρlist := RootsMultiplicityParametric(parametric_curve, 2);

  for i to nops(τlist) do
    τ := τlist[i, 1];
    n1 := τlist[i, 2];
    aux := factor(subs(s=τ, denom(p2)));
    if aux ≠ 0 then # Horizontal asymptote
      parametric_asymptote := [t, subs(s=τ, p2)];
    else # Parametrization asymptote
      for j from 1 to nops(ρlist) do
        ρ := ρlist[j, 1];
        if τ=ρ then
          break;
        end if;
      end do;
      n2 := ρlist[j, 2];
      f[n2] := simplify( $\frac{p2}{n2}, radical, symbolic$ );
      a[n2] := limit(f[n2], s=τ);
      tilde_r := a[n2]·tn2;
      for j from n2 - 1 by -1 to 0 do
        f[j] := simplify( $\frac{1}{p1^{n1}} \cdot (f[j+1] - a[j+1])$ , radical, symbolic);
        a[j] := (limit(f[j], s=τ));
        tilde_r := tilde_r + a[j]·tj;
      end do;
      parametric_asymptote := [tn1, tilde_r];
    end if;
    parametric_asymptoteslist := parametric_asymptote, parametric_asymptoteslist;
  end do;
  for j from 1 to nops(ρlist) do # Vertical asymptote
    ρ := ρlist[j, 1];
    aux := factor(subs(s=ρ, denom(p1)));
    if aux ≠ 0 then
      parametric_asymptote := [subs(s=ρ, p1), t];
      parametric_asymptoteslist := parametric_asymptote, parametric_asymptoteslist;
    end if;
  end do;

  return [parametric_asymptoteslist];
end proc;

```

## Example

```

> P := [ (s^3 + 2 s - 1) / ((s^2 - 1) * (s - 2)^2), 2 s^3 + s^2 + 1 ] / ((s - 2) * (s^2 - 1));
parametric_asymptotesL := AsymptotesParametricLimits(P);

parametric_asymptotesL := [ [t, 2/3], [t, -2 t + 3], [t^2, 7*sqrt(33)/11 t + 7/33] ]

```

## Procedure AsymptotesParametricDerivatives (parametric\_curve)

## Description

Computes the parametrization of the g-asymptotes of a given parametric curve using derivatives.

@param [{expression}, {expression}] parametric\_curve  
 @return [{expression}, {expression}], ... parametric\_asymptoteslist

## Code

```

> AsymptotesParametricDerivatives := proc(parametric_curve)
  local s, tau, n, i := 1, j;
  local q, q1, q2, sigma, rho, a, a_aux, tilde_r;
  local P, PP_aux, PP, overline_q;
  local rlist, rlist, alist, parametric_asymptoteslist := NULL;

  P := parametric_curve;
  s := indets(P)[1];
  PP_aux := [P[1], P[2], 1]; # Projective curve
  q := lcm(denom(P[1]), denom(P[2]));
  q1 := P[1] * q;
  q2 := P[2] * q;
  PP := [q1, q2, q];
  rlist := RootsMultiplicityParametric(P, 1);
  sigma := normal(q2 / q1);

  for i to nops(rlist) do
    tau := rlist[i, 1];
    n := rlist[i, 2];
    overline_q := radnormal(q / (s - tau)^n);
    rho := (q1 / overline_q)^(1/n);
    a := subs(s = tau, sigma);
    tilde_r := a * tau^n;
    for j from n - 1 by -1 to 0 do
      a_aux := normal(diff(diff(sigma, s) * rho^(n - j), [s$(n - 1 - j)]));
      a := 1 / (n - j)! * subs(s = tau, a_aux);
      tilde_r := tilde_r + a * tau^j;
    end do;
    parametric_asymptoteslist := [tau^n, tilde_r], parametric_asymptoteslist;
  end do;

  return [parametric_asymptoteslist];
end proc;

```

## Example

```

> P := [ (s^3 + 2 s - 1) / ((s^2 - 1) * (s - 2)^2), 2 s^3 + s^2 + 1 ] / ((s - 2) * (s^2 - 1));
parametric_asymptotesL := AsymptotesParametricDerivatives(P);

P := [ (s^3 + 2 s - 1) / ((s^2 - 1) * (s - 2)^2), 2 s^3 + s^2 + 1 ] / ((s - 2) * (s^2 - 1));
parametric_asymptotesL := [ [t, 2/3], [t, -2 t + 3], [t^2, 7*sqrt(11)*sqrt(3)/11 t + 7/33] ]

```

## References

1. Maxwell, E. *An Analytical Calculus*; Cambridge University Press: Cambridge, UK, 1962; Volume 3.
2. Zeng, G. Computing the asymptotes for a real plane algebraic curve. *J. Algebra* **2007**, *316*, 680–705. [[CrossRef](#)]
3. Bajaj, C.L. (Ed.) *Algebraic Geometry and its Applications*; Springer: Berlin/Heidelberg, Germany, 1994.
4. Cox, D.A.; Sturmfels, B. Applications of Computational Algebraic Geometry. In *Proceedings of the Symposia in Applied Mathematics*; American Mathematical Society: Providence, RI, USA, 1998; Volume 53.
5. Hoffmann, C.M.; Sendra, J.R.; Winkler, F. Parametric algebraic curves and applications. *J. Symb. Comput.* **1997**, *23*. Available online: <https://dl.acm.org/toc/josc/1997/23/2-3> (accessed on 1 January 2022).
6. Schicho, J.; Sendra, J.R. Special Issue on Algebraic Curves. *Appl. Algebra Eng. Commun. Comput.* **2007**, *18*, 1–2. [[CrossRef](#)]
7. Poulakis, D.; Voskos, E. On the Practical Solution of Genus Zero Diophantine Equations. *J. Symb. Comput.* **2000**, *30*, 573–582. [[CrossRef](#)]
8. Poulakis, D.; Voskos, E. Solving Genus Zero Diophantine Equations with at Most Two Infinite Valuations. *J. Symb. Comput.* **2002**, *33*, 479–491. [[CrossRef](#)]
9. Bajaj, C.; Lee, H.Y.; Merkert, R.; Pascucci, V. NURBS Based B-Rep Models for Macromolecules and their Properties. In *Proceedings of the Fourth ACM Symposium on Solid Modeling and Applications*, Atlanta, GA, USA, 14–16 May 1997; pp. 217–228. [[CrossRef](#)]
10. Goppa, V.D. Codes associated with divisors. *Probl. Peredachi Informatsii* **1977**, *13*, 33–39.
11. Goppa, V.D. Codes on algebraic curves. *Sov. Math. Dokl.* **1981**, *259*, 1289–1290.
12. Pretzel, O. *Codes and Algebraic Curves*; Oxford Lecture Series in Mathematics and its Applications, 8; Oxford University Press: Oxford, UK, 1998.
13. Abbaszadeh Sori, A.; Ebrahimnejad, A.; Motameni, H. Elite artificial bees' colony algorithm to solve robot's fuzzy constrained routing problem. *Comput. Intell.* **2020**, *36*, 659–681. [[CrossRef](#)]
14. Blake, I.F.; Seroussi, G.; Smart, N.P. *Elliptic Curves in Cryptography*; London Mathematical Society Lecture Note Series, 265; Cambridge University Press: Cambridge, UK, 1999.
15. Buchmann, J.A. *Introduction to Cryptography*; Undergraduate Texts in Mathematics; Springer: New York, NY, USA, 1995.
16. Koblitz, N. *Algebraic Aspects of Cryptography*; Algorithms and Computation in Mathematics; Springer: Berlin/Heidelberg, Germany, 1998.
17. Koblitz, N. Good and bad uses of elliptic curves in cryptography. *Mosc. Math. J.* **2002**, *2*, 693–715. [[CrossRef](#)]
18. Farin, G. *Curves and Surfaces for Computer-Aided Geometric Design*, 3rd ed.; Academic Press: Boston, MA, USA, 1993.
19. Farin, G.; Hoschek, J.; Kim, M.S. *Handbook of Computer Aided Geometric Design*; North-Holland: Amsterdam, The Netherlands, 2002.
20. Hoschek, J.; Lasser, D. *Fundamentals of Computer-Aided Geometric Design*; A. K. Peters/CRC Press: Wellesley, MA, USA, 1993.
21. Sederberg, T.W. Applications to Computer Aided Geometric Design. In *Proceedings of the Symposia in Applied Mathematics: Applications of Computational Algebraic Geometry*; Cox, D.A., Sturmfels, B., Eds.; American Mathematical Society: Providence, RI, USA, 1998; Volume 53, pp. 67–89. [[CrossRef](#)]
22. Götz, M.; Maymeskul, V.; Saff, E. Asymptotic Distribution of Nodes for Near-Optimal Polynomial Interpolation on Certain Curves in  $\mathbb{R}^2$ . *Constr. Approx.* **2002**, *18*, 255–284. [[CrossRef](#)]
23. Forsman, K. On Rational State Space Realizations. In *Nonlinear Control Systems Design: Selected Papers from the 2nd IFAC Symposium*; Fliess, M., Ed.; IFAC Symposia Series; Pergamon Press: Oxford, UK, 1993; pp. 341–346.
24. Blasco, A.; Pérez-Díaz, S. Asymptotic behavior of an implicit algebraic plane curve. *Comput. Aided Geom. Des.* **2014**, *31*, 345–357. [[CrossRef](#)]
25. Campo-Montalvo, E.; Fernández de Sevilla, M.; Pérez-Díaz, S. A simple formula for the computation of branches and asymptotes of curves and some applications. *Comput. Aided Geom. Des.* **2022**, *94*, 102084. [[CrossRef](#)]
26. Blasco, A.; Pérez-Díaz, S. A new approach for computing the asymptotes of a parametric curve. *J. Comput. Appl. Math.* **2020**, *364*, 112350. [[CrossRef](#)]
27. Shen, L.; Chionh, E.; Gao, X.S.; Li, J. Proper reparametrization for inherently improper unirational varieties. *J. Syst. Sci. Complex.* **2010**, *24*, 367–380. [[CrossRef](#)]
28. Abkar, A.; Shekarbaigi, M. A Novel Iterative Algorithm Applied to Totally Asymptotically Nonexpansive Mappings in CAT(0) Spaces. *Mathematics* **2017**, *5*, 14. [[CrossRef](#)]
29. Paik, J.W.; Lee, K.H.; Lee, J.H. Asymptotic Performance Analysis of Maximum Likelihood Algorithm for Direction-of-Arrival Estimation: Explicit Expression of Estimation Error and Mean Square Error. *Appl. Sci.* **2020**, *10*, 2415. [[CrossRef](#)]
30. Jeong, S.H.; Son, B.k.; Lee, J.H. Asymptotic Performance Analysis of the MUSIC Algorithm for Direction-of-Arrival Estimation. *Appl. Sci.* **2020**, *10*, 2063. [[CrossRef](#)]
31. Harbau, M.H.; Ugwunnadi, G.C.; Jolaoso, L.O.; Abdulwahab, A. Inertial Accelerated Algorithm for Fixed Point of Asymptotically Nonexpansive Mapping in Real Uniformly Convex Banach Spaces. *Axioms* **2021**, *10*, 147. [[CrossRef](#)]
32. Sendra, J.R.; Winkler, F.; Pérez-Díaz, S. *Rational Algebraic Curves: A Computer Algebra Approach*; Algorithms and Computation in Mathematics; Springer: Berlin/Heidelberg, Germany, 2007. [[CrossRef](#)]
33. Khan, M.; Santos-García, G.; Noor, M.; Soliman, M. Some new concepts related to fuzzy fractional calculus for up and down convex fuzzy number valued functions and inequalities. *Chaos Solut. Fractals* **2022**, *164*, 112692. [[CrossRef](#)]

34. Di Caprio, D.; Ebrahimnejad, A.; Alrezaamiri, H.; Santos-Arteaga, F.J. A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights. *Alex. Eng. J.* **2022**, *61*, 3403–3415. [[CrossRef](#)]
35. Ebrahimnejad, A.; Karimnejad, Z.; Alrezaamiri, H. Particle swarm optimisation algorithm for solving shortest path problems with mixed fuzzy arc weights. *Int. J. Appl. Decis. Sci.* **2015**, *8*, 203–222. [[CrossRef](#)]
36. Poteaux, A.; Rybowicz, M. Improving Complexity Bounds for the Computation of Puiseux Series over Finite Fields. In Proceedings of the 2015 ACM on International Symposium on Symbolic and Algebraic Computation, Bath, UK, 6–9 July 2015; pp. 299–306. [[CrossRef](#)]
37. Baur, W.; Strassen, V. The complexity of partial derivatives. *Theor. Comput. Sci.* **1983**, *22*, 317–330. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.