Annals of Mathematics and Artificial Intelligence (2023) 91:537–561
https://doi.org/10.1007/s10472-023-09856-z

Design and implementation of symbolic algorithms for the computation of generalized asymptotes

Marián Fernández de Sevilla[1] · **Rafael Magdalena Benedicto**[2] · **Sonia Pérez-Díaz**[3]

Accepted: 1 May 2023 / Published online: 3 July 2023

Abstract
In this paper we present two algorithms for computing the *g-asymptotes* or *generalized asymptotes*, of a plane algebraic curve, $\mathscr{C}$, implicitly or parametrically defined. The asymptotes of a curve $\mathscr{C}$ reflect the status of $\mathscr{C}$ at points with sufficiently large coordinates. It is well known that an asymptote of a curve $\mathscr{C}$ is a line such that the distance between $\mathscr{C}$ and the line approaches zero as they tend to infinity. However, a curve $\mathscr{C}$ may have more general curves than lines describing the status of $\mathscr{C}$ at infinity. These curves are known as *g-asymptotes* or *generalized asymptotes*. The pseudocodes of these algorithms are presented, as well as the corresponding implementations. For this purpose, we use the algebra software `Maple`. A comparative analysis of the algorithms is carried out, based on some properties of the input curves and their results to analyze the efficiency of the algorithms and to establish comparative criteria. The results presented in this paper are a starting point to generalize this study to surfaces or to curves defined by a non-rational parametrization, as well as to improve the efficiency of the algorithms. Additionally, the methods developed can provide a new and different approach in prediction (regression) or classification algorithms in the machine learning field.

Keywords Algebraic curves · Infinity branches · Convergent branches · Approaching curves · Generalized asymptotes · Algorithm performance · Hardware usage

Sonia Pérez-Díaz, Marián Fernández de Sevilla and Rafael Magdalena Benedicto are contributed equally to this work.

✉ Sonia Pérez-Díaz
sonia.perez@uah.es

Marián Fernández de Sevilla
marian.fernandez@uah.es

Rafael Magdalena Benedicto
rafael.magdalena@uv.es

[1] University of Alcalá, Department of Computer Science, Ctra. Madrid-Barcelona, km.33,6, Alcalá de Henares 28871, Madrid, Spain

[2] Department of Electronic Engineering, University of Valencia, Av. Universidades s/n, Burjassot 46100, Valencia, Spain

[3] University of Alcalá, Department of Physics and Mathematics, Ctra. Madrid-Barcelona, km.33,6, Alcalá de Henares 28871, Madrid, Spain

Mathematics Subject Classification (2010) 14Q05 · 51N35 · 70G55 · 94B27

1 Introduction

In this paper, we present the design and implementation of two `Maple` packages to deal with some algebraic-geometric constructions with curves that appear in practical applications in Computer Aided Design.

More precisely, we develop an algorithmic solution that allows determining the behavior at infinity of a plane algebraic curve, $\mathscr{C}$, by determining the *generalized asymptotes*, or *g-asymptotes*, of its branches at points with sufficiently large coordinates, i.e., at the infinity points. The notion of *g-asymptote* generalizes the classical concept of (line) asymptote and its calculation methods (see [12, 16]).

This question is very important in the study of algebraic curves because the asymptotes provide information about the behavior of curves at infinity. For example, determining the asymptotes of a curve is essential to drawing its graph.

The generalized asymptotes

The asymptotes of an infinity branch, B, of a real plane algebraic curve, $\mathscr{C}$, determine the behavior of B at the points with "large coordinates". In analytic geometry, an asymptote of a curve is a line such that the distance between the curve and the line approaches zero as they tend to infinity. In the field of algebraic geometry, an asymptote is a tangent line to the curve at the infinity.

If B can be explicitly defined as $y = f(x)$ or $x = g(y)$ (f and g are continuous functions on an infinite interval), one may easily decide whether $\mathscr{C}$ has an asymptote at B by analyzing the existence of the limits of certain functions when x (or y) tends to ∞. In fact, if these limits can be computed, we may obtain the equation of the asymptote of $\mathscr{C}$ at B. However, if this branch B cannot be defined in an explicit way, both the decision and the computation of the asymptote of $\mathscr{C}$ at B, require some other tools.

It is well known that an algebraic curve may have more general curves than lines describing the status of a branch at the points with sufficiently large coordinates. In this sense, we say that a curve $\widetilde{\mathscr{C}}$ is a *generalized asymptote* (or *g-asymptote*) of a given curve $\mathscr{C}$ if the distance between $\widetilde{\mathscr{C}}$ and $\mathscr{C}$ tends to zero as they tend to infinity, and $\mathscr{C}$ cannot be approached by another curve of lower degree (see [1–3]). This motivates our interest in efficiently computing these *generalized asymptotes*.

An example is illustrated in Fig. 1 where it can be seen that a hyperbola is a degree 2 curve with two real asymptotes, which implies that the hyperbola degenerates at infinity, into two lines. The behavior of an ellipse is similar, although in this case, the infinity branches are complex and therefore, the ellipse degenerates at infinity into two complex lines.

However, the asymptotic behavior of a parabola is different, since at infinity the parabola cannot be approached by any straight line, that is, by any curve of smaller degree (see Fig. 2).

As we said previously, the concept of g-asymptote is similar to the classic concept of asymptote [10, 12, 14]. The difference is that a g-asymptote is not necessarily a line, but a higher degree curve of the smallest possible degree that approaches the curve at infinity (a *perfect curve*). See Section 2 for the formal definition. To clarify this notion, we next consider a plane curve $\mathscr{C}$ defined by the irreducible polynomial $f(x, y) = -yx - y^2 - x^3 + 2x^2y + x^2 - 2y \in \mathbb{R}[x, y]$. The given curve $\mathscr{C}$ has degree 3 and two infinity branches. In

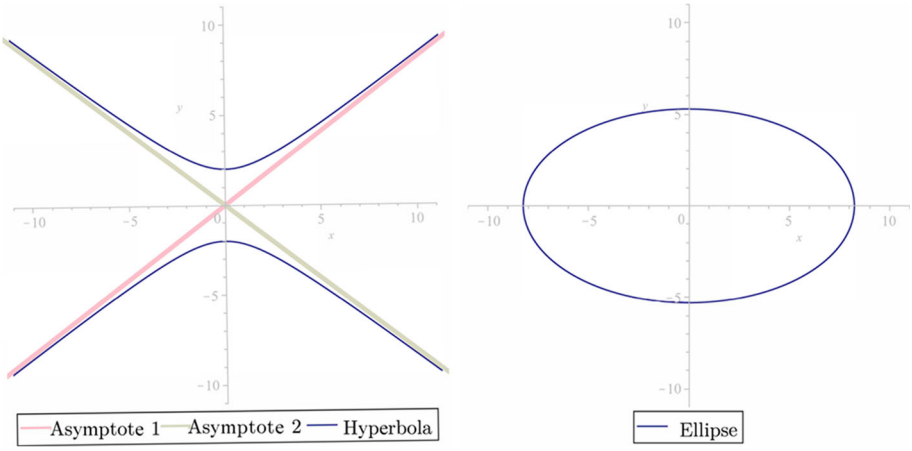

Fig. 1 Asymptotic behavior of the hyperbola (left) and ellipse (right).

Fig. 3, it can be seen that these infinity branches are approached by the parabola defined by the polynomial $y - 2x^2 + 3/2x + 15/8$ and by the line defined by $y - x/2 + 1/8$.

Other plane curves of degree 3, such as $y - x^3 = 0$ or $y^2 - x^3 = 0$, cannot be approached by any curve of degree less than 3.

Artificial intelligence applications

We strongly believe in the possibility of applying the method presented in this paper to the field of artificial intelligence (AI) and specifically in the area of machine learning (ML).

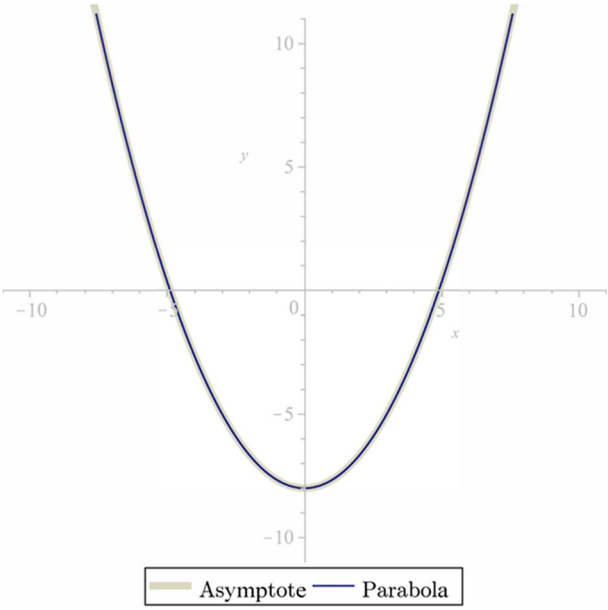

Fig. 2 Asymptotic behavior of the parabola.

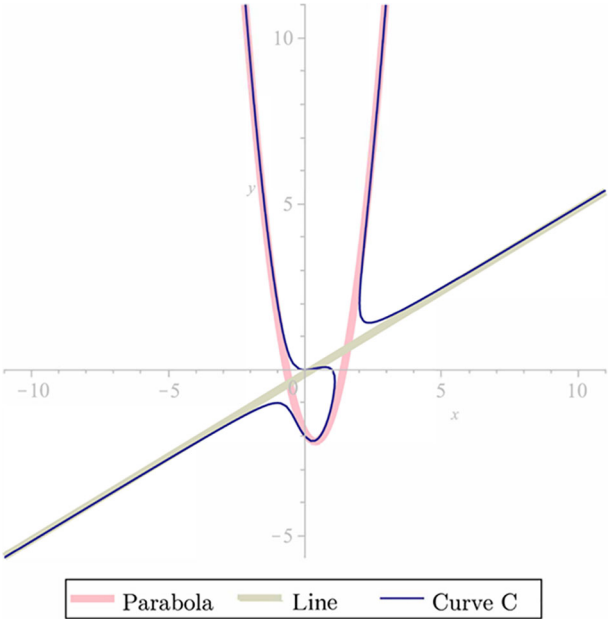

Fig. 3 Curve $\mathscr{C}$ approached by a parabola and a straight line

These symbolic computing tools can provide new solutions to the analysis of data patterns, offering an alternative solution to the traditional identification techniques, based on symbolic reasoning, providing simple and efficient methods for systems that are difficult to observe or where mathematics are very complicated ([5]).

Future work

We should emphasize that the proposal presented in the paper is opening a promising line of research in a different version of several ML algorithms. First, the methods proposed in the paper can be easily generalized to high dimensional spaces (see [3] and [7]), enabling for separating different classes using generalized asymptotic hyperplanes in classification problems. Besides, a piecewise generalized asymptotic class separation could be an extension of the proposed future work that enables a piecewise hyper dimensional generalized asymptote [8]. Piecewise points should be determined based on data distribution and singular areas, determined by data distribution. The concept of g-asymptote in the n-dimensional space could thus be applied to extend the clustering on the whole space.

Finally, Artificial Neural Networks (ANN), Deep Learning methods and other algorithms that iteratively minimize a cost function for updating the weights could be an interesting field of extending the purpose of this paper, considering that weights values tend asymptotically to the optimal values that minimize the error or cost function [9, 11]. The evolution of the weights updates themselves could be characterized and asymptotically projected. The study of the behavior when iterations grow, and the algebraic outcome of an asymptotic weight convergence function could bring some new trends on weight optimization or interpretability of the trained networks.

Structure of the paper

In order to develop the algorithms and to describe the packages, we first recall the basic required mathematical background. Thus, Section 2 presents the main notions and explains the intuitive idea of *perfect curve* and *g-asymptote*. That is, given a curve $\mathscr{C}$, a curve $\widetilde{\mathscr{C}}$ is said to be a *g-asymptote* of $\mathscr{C}$ if $\widetilde{\mathscr{C}}$ is a curve of the smallest possible degree that approaches $\mathscr{C}$ at infinity (see [1, 2]).

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 have been illustrated with examples that show the methods developed in this work, as well as the implementation programmed with the algebra system `Maple`, which has been included in Appendix B.

Section 4 analyzes the performance of the previous algorithms based on the computation of the Puiseux series. For this purpose, we study several cases to observe the system overload, when Algorithms 3.1 or 3.2 are executed.

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, we analyze the efficiency of each algorithm and we establish a comparative criteria regarding the time of use of the CPU from the simplest curve to the curve with the highest complexity.

2 Generalized asymptotes of algebraic curves

This section starts by introducing the notions of *infinity branches*, *convergent branches* and *approaching curves*, derived from previous research (see [1, 2, 6, 7]).

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

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

with $d := \deg(\mathscr{C})$, and $f_i(x, y)$ the homogeneous form of degree i, for $i = 0, \ldots, d$. Let $(1 : m : 0)$, $m \in \mathbb{C}$ be the infinity points of $\mathscr{C}^*$ (if $(0 : 1 : 0)$ is an infinity point of the input curve, a linear change of coordinates must be applied).

Under these conditions and in order to get the infinity branches of $\mathscr{C}$, we consider the curve defined by the polynomial $g(y, z) = F(1, y, z)$, and we compute the series expansion for the solutions of $g(y, z) = 0$ around $z = 0$. There exist exactly $\deg_y(g)$ solutions given by different Puiseux series φ_i, $i = 1 \ldots \deg_y(g)$. In the following, we denote as

$$\varphi(t) = m + a_1 t^{N_1/N} + a_2 t^{N_2/N} + a_3 t^{N_3/N} + \cdots \in \mathbb{C} \ll t \gg, \quad a_i \neq 0, \forall i \in \mathbb{N},$$

where $\mathbb{C} \ll t \gg$ is the field of *formal Puiseux series*, $N_i \in \mathbb{N}$, $i = 1, \ldots$, and $0 < N_1 < N_2 < \cdots$ one of these series. Therefore, $g(\varphi(t), t) = 0$ in a neighborhood of $t = 0$ where $\varphi(t)$ converges.

From these Puiseux series, we get the following definition of *infinity branch* of $\mathscr{C}$ (see [2]).

Definition 1 An *infinity branch* of a plane algebraic curve $\mathscr{C}$, 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 $r(z) = z\varphi_i(z^{-1}) = mz + a_1 z^{1-N_1/N} + a_2 z^{1-N_2/N} + a_3 z^{1-N_3/N} + \cdots$ and M is some natural number.

In the following, we introduce the notions of convergent branches and approaching curves. Intuitively speaking, two infinity branches converge if they get closer as they tend to infinity. This concept allows us to analyze whether two curves approach each other.

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

Definition 3 Let $\mathscr{C}$ be a plane algebraic curve with an infinity branch B. We say that a curve $\overline{\mathscr{C}}$ *approaches* $\mathscr{C}$ at the branch B, if $\lim_{z\to\infty} d((z, r(z)), \overline{\mathscr{C}}) = 0$, where $d(p, \overline{\mathscr{C}}) = \min\{d(p, q) : q \in \overline{\mathscr{C}}\}$, and $d(p, q)$ denotes the Euclidean distance between the points p and q.

In [2], we show that if $\mathscr{C}$ is a plane curve with an infinity branch B, then a plane curve $\overline{\mathscr{C}}$ *approaches* $\mathscr{C}$ into B, if and only if $\overline{\mathscr{C}}$ has an infinity branch $\overline{B}$ such that B and $\overline{B}$ are convergent.

Given a plane curve $\mathscr{C}$ and an infinity branch B, we have described how $\mathscr{C}$ can be approached at B by a second curve $\overline{\mathscr{C}}$. Let us suppose that $\deg(\overline{\mathscr{C}}) < \deg(\mathscr{C})$. Then, one may say that $\mathscr{C}$ degenerates, since it behaves at infinity as a curve of smaller degree. As we said in the introduction, a hyperbola is a curve of degree 2 that has two real asymptotes, which implies that the hyperbola degenerates, at infinity, to two lines (see Fig. 1). However, as we said, the asymptotic behavior of a parabola is different, since it cannot be approached at infinity by any line (see Fig. 2). This motivates the definitions of *perfect curve* and *generalized asymptote* (see [1]).

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

A curve $\mathscr{C}$ that is not perfect can be approached by other curves of smaller degree. If these curves are perfect, we call them *g-asymptotes*, and we represent it as $\widetilde{\mathscr{C}}$. More precisely, we have the following definition.

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

We emphasize that the notion of g-asymptote is similar to the classical concept of asymptote. The difference is that a g-asymptote is not necessarily a line, but a perfect curve. Actually, it is a generalization, since every line is a perfect curve. Throughout the paper, we sometimes refer to g-asymptote simply as asymptote

3 Algorithms for computing generalized asymptotes

In this section, we describe two algorithms that construct the parametrizations of the generalized asymptotes of the infinity branches of a curve $\mathscr{C}$; both are described by the corresponding pseudocode.

First, in Section 3.1, we introduce Algorithm 3.1 which computes the g-asymptotes of a curve given by its implicit equation. In Section 3.2, we present a new algorithm, Algorithm 3.2, which is applied to a parametrically given curve.

3.1 Algorithm for curves implicitly defined

Let $\mathscr{C}$ be a curve with a branch

$$B = \{(z, r(z)) \in \mathbb{C}^2 : z \in \mathbb{C},\ \|z\| > M\},$$

where $r(z) = mz + a_1 z^{1-N_1/N} + \cdots + a_k z^{1-N_k/N} + a_{k+1} z^{1-N_{k+1}/N} + \cdots$, with coefficients $a_1, a_2, \ldots \in \mathbb{C} \setminus \{0\}$, $m \in \mathbb{C}$, $N, N_1, N_2 \ldots \in \mathbb{N}$, and $0 < N_1 < N_2 < \cdots$. Suppose that $N_k \leq N < N_{k+1}$, i.e., the monomials $a_j z^{1-N_j/N}$ with $j \geq k+1$ have a negative exponent. In the following, we write

$$r(z) = mz + a_1 z^{1-n_1/n} + \cdots + a_k z^{1-n_k/n} + a_{k+1} z^{1-N_{k+1}/N} + \cdots$$

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

$$\widetilde{r}(z) = mz + a_1 z^{1-n_1/n} + \cdots + a_k z^{1-n_k/n}. \tag{1}$$

Applying the change $z = t^n$, we obtain a proper (or birational) parametrization of a curve $\widetilde{\mathscr{C}}$

$$\widetilde{\mathscr{P}}(t) = (t^n, mt^n + a_1 t^{n-n_1} + \cdots + a_k t^{n-n_k}) \in \mathbb{C}[t]^2, \tag{2}$$

where $n, n_1, \ldots, n_k \in \mathbb{N}$, $\gcd(n, n_1, \ldots, n_k) = 1$, and $0 < n_1 < \cdots < n_k$, which is an asymptote of $\mathscr{C}$ (see Lemma 3 and Theorem 2 [1]).

The notion of proper parametrization can be revised in [15] (see Section 4.1 in Chapter 4). We remind that the question of determining a birational (proper) reparametrization of an input non-birational parametrization has been analyzed by several authors and there are effective answers to approach it (see Section 4.2 in [15]).

Algorithm 3.1 computes the parametrizations of the asymptotes of the infinity branches of the curve $\mathscr{C}$ which is implicitly defined. In Example 1, we illustrate this algorithm, which has been implemented with the mathematical software `Maple` (see Appendix B).

Algorithm 1 Computation of asymptotes of an implicit curve $\mathscr{C}$.

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

We remark that the routine $ProjectiveCurve(\mathscr{C})$ determines the projective curve associated to the affine curve $\mathscr{C}$ which is defined by the homogenization, $F(x, y, z)$, of the polynomial $f(x, y)$ that defines $\mathscr{C}$. The routine $InfinityPoints(F(x, y, 0))$ returns the infinity points of the projective curve. The routine *PuiseuxSeries*$(g(y, z),\ z = 0,\ y)$ computes the Puiseux Series of the polynomial $g(y, z)$ around $z = 0$.

Example 1 Let $\mathscr{C}$ 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) $\tilde{r}_1(z) = -\dfrac{48^{1/5}}{2}z.$ b) $\widetilde{\mathscr{P}}_1(t) = \left(t^5, -\dfrac{48^{1/5}}{2}t\right).$

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) $\tilde{r}_2(z) = -2z.$ b) $\widetilde{\mathscr{P}}_2(t) = (t, -2t).$

Figure 4 plots the curve $\mathscr{C}$ and its generalized asymptotes $\widetilde{\mathscr{C}_1}$ and $\widetilde{\mathscr{C}_2}$, defined by the parametrizations $\widetilde{\mathscr{P}}_1(t)$ and $\widetilde{\mathscr{P}}_2(t)$, respectively.

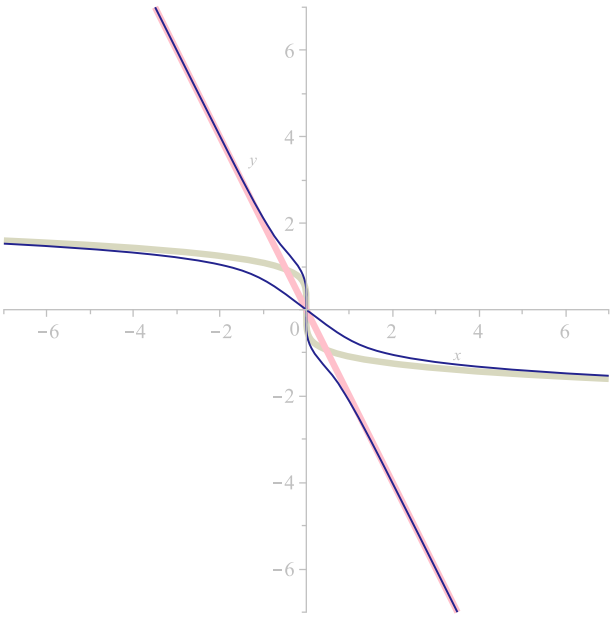

Fig. 4 Infinity asymptotes $\widetilde{\mathscr{C}_1}$ (beige) and $\widetilde{\mathscr{C}_2}$ (pink) of the curve $\mathscr{C}$.

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 (see [3, Sec.5]). This method also involves the computation of Puiseux series and infinity branches (see [4]).

Let $\mathscr{C}$ be a plane curve defined by the parametrization

$$\mathscr{P}(s) = (p_1(s), p_2(s)) \in \mathbb{R}(s)^2, \ p_i(s) = p_{i1}(s)/p_{i2}(s), \tag{3}$$

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

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

$$\mathscr{P}^*(s) = \left(1 : \frac{p_2(s)}{p_1(s)} : \frac{1}{p_1(s)}\right). \tag{4}$$

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

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

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

For this purpose, 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 $\mathscr{C}^*$. Observe that, in this section, we are given the parametrization $\mathscr{P}^*(s)$ of $\mathscr{C}^*$ and then,

$$F(\mathscr{P}^*(s)) = F\,(1 : p_2(s)/p_1(s) : 1/p_1(s)) = 0.$$

Thus, intuitively speaking, in order to compute the infinity branches of $\mathscr{C}$, and in particular the series φ, one needs to rewrite the parametrization $\mathscr{P}^*(s)$ 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 $\mathscr{P}^*(\ell(t)) = (1 : \varphi(t) : t)$ around $t = 0$.

Hence, from the above reasoning, we deduce that first, we have to consider the equation

$$1/p_1(s) = t, \quad \text{or equivalently,} \quad p_{12}(s) - tp_{11}(s) = 0 \tag{5}$$

and we solve it in the variable s around $t = 0$. From Puiseux's Theorem, there exist solutions $\ell_1(t), \ell_2(t), \ldots, \ell_k(t) \in \mathbb{C} \ll t \gg$ such that, $p_{12}(\ell_i(t)) - tp_{11}(\ell_i(t)) = 0, i \in \{1, \ldots, k\}$, in a neighborhood of $t = 0$.

Thus, for each $i \in \{1, \ldots, k\}$, 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

$$\varphi_i(t) = \frac{p_2(\ell_i(t))}{p_1(\ell_i(t))}, \tag{6}$$

are in $\mathscr{C}^*$ for $\|t\| < M_i$ (note that $\mathscr{P}^*(\ell(t)) \in \mathscr{C}^*$, since $\mathscr{P}^*(\ell(t))$ is a parametrization of $\mathscr{C}^*$). 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 $\mathscr{C}$ for $\|z\| > M_i^{-1}$. Hence, the infinity branches of $\mathscr{C}$ are the sets $B_i = \{(z, r_i(z)) \in \mathbb{C}^3 : z \in \mathbb{C},\ \|z\| > M_i^{-1}\},\ i \in \{1, \ldots, k\}$.

Remark 1 Note that the series $\ell_i(t)$ satisfies that $p_1(\ell_i(t))\, t = 1$, for $i \in \{1, \ldots, k\}$. Then, from equality (6), 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 monomials with negative exponent from $r_i(z)$.

The following algorithm computes the infinity branches of a given parametric curve, and it provides a g-asymptote for each of the infinity branch. Similarly, as in Algorithm 3.1, the routine $ProjectiveCurve(\mathscr{C})$ determines the projective curve associated to the affine curve $\mathscr{C}$ which is defined by the homogenization, $F(x, y, z)$, of the polynomial $f(x, y)$ that defines $\mathscr{C}$. The routine $InfinityPoints(F(x, y, 0))$ returns the infinity points of the projective curve. The routine $PuiseuxSeries(g(y, z),\ z = 0,\ y)$ computes the Puiseux Series of the polynomial $g(y, z)$ around $z = 0$. Furthermore, the routine $ProjectiveParametrization(\mathscr{P}(s))$ determines the projective curve associated to the curve $\mathscr{C}$ which is defined by the parametrization $\mathscr{P}(s)$.

Algorithm 2 Computation of asymptotes of a parametric curve $\mathscr{C}$.

Require: $\mathscr{C}$, plane algebraic curve defined by
$\mathscr{P}(s) \leftarrow (p_1(s), p_2(s)) \in \mathbb{R}(s)^2,\ p_i(s) = p_{i1}(s)/p_{i2}(s)$, /* Equation 3 */
$\gcd(p_{i1}(s), p_{i2}(s)) = 1,\ i = 1, 2$
Ensure: $\mathscr{C}_i,\ i = 1, \ldots, k$ /* Asymptotes of $\mathscr{C}$ */
1: $\mathscr{P}^*(s) \leftarrow ProjectiveParametrization(\mathscr{P}(s))$ /* Equation 4 */
2: $P_1, \ldots P_m \leftarrow InfinityPoints(\mathscr{P}^*(s))$
3: $\ell_1(t), \ldots, \ell_k(t) \leftarrow PuiseuxSeries\,(p_{12}(s) - tp_{11}(s) = 0, t = 0, s)$/* Eq. 5 */
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: $\widetilde{r_i}(z) \leftarrow m_i z + a_{1,i} z^{1-n_{1,i}/n_i} + \cdots + a_{k_i,j} z^{1-n_{k_i,i}/n_i}$ /* Equation 1 */
9: $\widetilde{\mathscr{P}_i}(t) \leftarrow (t^{n_i}, \widetilde{r_i}(t^{n_i})) \in \mathbb{C}[t]^2,\ i = 1, \ldots, k$ /* Equation 2 */ X
10: **end for**
11: $\widetilde{\mathscr{C}_i} \leftarrow \widetilde{\mathscr{P}_i}(t) \in \mathbb{C}[t]^2, i = 1, \ldots, k$

It should be recalled that the above algorithm has been implemented in `Maple` (see Appendix B). We illustrate it by the following example.

Example 2 Let $\mathscr{C}$ be the plane curve defined by

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

We apply Algorithm 3.2 to compute the asymptotes of $\mathscr{C}$. For this purpose, we start computing the Puiseux solutions of the equation $p_{12}(s) - tp_{11}(s) = 0$ around $t = 0$. We get:

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

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

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

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) $\widetilde{r}_1(z) = \frac{2}{5}z - \frac{3}{4}$. b) $\widetilde{\mathscr{P}}_1(t) = \left(t,\ \frac{2}{5}t - \frac{3}{4}\right)$.

Iteration 2:

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

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) $\widetilde{r}_3(z) = -5$. b) $\widetilde{\mathscr{P}}_3(t) = (t,\ -5)$.

The asymptotes are the plane curves $\widetilde{\mathscr{C}}_1$, $\widetilde{\mathscr{C}}_2$ and $\widetilde{\mathscr{C}}_3$ defined by the proper rational parametrizations, $\widetilde{\mathscr{P}}_1(t)$, $\widetilde{\mathscr{P}}_2(t)$ and $\widetilde{\mathscr{P}}_3(t)$ (see Fig. 5).

4 Symbolic algorithms for the computation of generalized asymptotes

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 for its implicit expression, considering the infinity branches that converge to the given curve. The second method deals with the case of parametrically defined curves. Both algorithms are based on the computation of the Puiseux series.

For this purpose, fourteen study cases have been selected as input for the implementation of Algorithms 3.1 and 3.2. The execution has allowed us to observe the time and memory usage information for executing a process, obtaining the system overload, and the computational performance.

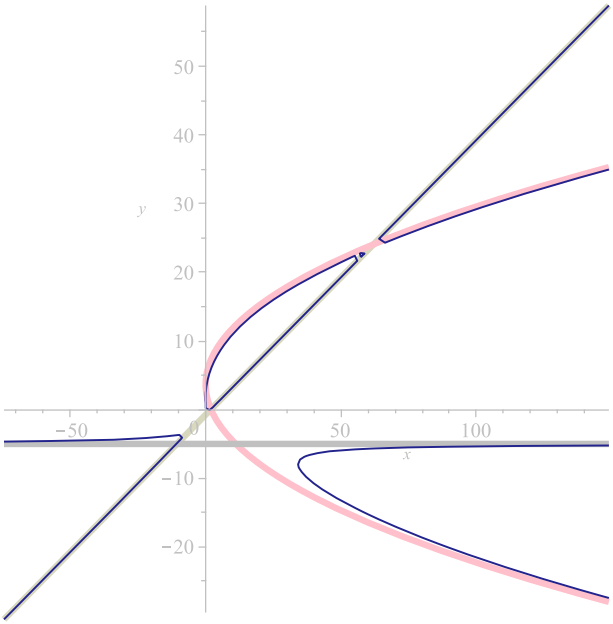

Fig. 5 Infinity asymptotes $\widetilde{\mathscr{C}_1}$ (beige), $\widetilde{\mathscr{C}_2}$ (pink) and $\widetilde{\mathscr{C}_3}$ (grey) of the curve $\mathscr{C}$.

4.1 Analysis of the computational performance

This subsection analyzes the computational performance of the algorithms defined in the previous section, when the procedures implemented in the Appendix B are executed. Thus, the usage of CPU and memory, as well as the real time that the process remains in the system, are evaluated.

In order to measure the efficiency of the algorithms, one hundred input parametric curves were randomly generated. For this purpose, we use the command `randpoly(vars, opts)` of `Maple` to generate random polynomials, $p_i, q_i,\ i = 1, 2$ in *vars* (in this case in the variable t). These polynomials will be used for defining the rational parametrization $\mathscr{P}(t) = (p_1(t)/q_1(t), p_2(t)/q_2(t))$.

We remind that once we have the input parametrization, we consider a change of variables so that the degree of the denominator is equal or greater than the degree of the numerator and $(0 : 1 : 0)$ is not an infinity point. We recall that we can compute the implicit polynomial defining these curves by applying for instance resultants (see [15]). Under these conditions, seven classes were created according to the following properties:

1. Degree of the curve.
2. Number of monomials.

Subsequently, two curves from each group were randomly chosen, obtaining a set of fourteen curves with different characterizations (see Appendix A). All these curves determine the study cases to analyze the computational performance of the algorithms presented in Section 3, which have been programmed with the algebra system `Maple`, according to the implementation presented in Appendix B.

Based on all the above considerations, an analysis of the computational performance of each one of the algorithms presented has been carried out. Thus, the hardware overload

degree of the machine resources (CPU and memory) has been 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 B. In addition, the value of real time invested by the machine has been calculated for both methods presented in Section 3.

To quantify these results, it has been necessary to use the tools provided by `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 mono-processor 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 the precise overload degree, each algorithm under analysis has been iterated one hundred times for each one of the study curves. Consequently, it has been possible to record reliable results on the system requirements, 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 monomials, as well as the following results obtained after each execution:

1. Number of infinity branches.
2. Highest degree of the analyzed asymptotes.
3. Number of real asymptotes.
4. Number of complex asymptotes.

It is important to note that Algorithms 3.1 and 3.2 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 these Puiseux series expansions.

From a hardware point of view, the processes have been 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 1,536 MiB Graphics 655, and 500 GiB SSD. The computer algebra software `Maple 2021.1` has been 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. To facilitate the reading of some expressions, certain results have been represented in floating-point.

4.1.1 Computation of asymptotes of implicit algebraic curves

This subsection presents the results of running Algorithm 3.1 one hundred times, with accuracy equal to ten for the Puiseux series expansion, on the fourteen curves in the Appendix A.

Table 1 shows the properties of the input curves and their asymptotes. The curve with the highest degree and the highest number of monomials is $\mathscr{C}_3$. One may check that the asymptotes of this curve have degrees 1 and 3, much smaller than the degree of the given input curve, whose value is 17.

Table 1 Properties of the implicit curves and their asymptotes

Id	Deg.	# Monom.	# Branch	MaxDeg	# Real Asymp.	# Complex Asymp.
$\mathscr{C}_1$	7	21	5	2	3	2
$\mathscr{C}_2$	7	21	3	3	3	–
$\mathscr{C}_3$	17	114	5	3	1	4
$\mathscr{C}_4$	3	8	2	2	2	–
$\mathscr{C}_5$	6	25	3	4	3	–
$\mathscr{C}_6$	4	11	3	2	3	–
$\mathscr{C}_7$	4	12	2	3	2	–
$\mathscr{C}_8$	5	13	2	3	2	–
$\mathscr{C}_9$	5	15	2	2	2	–
$\mathscr{C}_{10}$	7	19	3	3	1	2
$\mathscr{C}_{11}$	15	58	4	5	2	2
$\mathscr{C}_{12}$	13	31	7	2	1	6
$\mathscr{C}_{13}$	9	22	5	2	3	2
$\mathscr{C}_{14}$	5	15	2	2	2	–

More precisely, the parametric equations of the convergent asymptotes with the curve $\mathscr{C}_3$ are:

- $\widetilde{\mathscr{P}}_{3,1}(t) := \left(t^4, -\dfrac{731760468\ldots70022160}{(2941595+1044237\alpha)^9} - \dfrac{31007116\ldots24025488}{(2941595+1044237\alpha)^9}\,\alpha\right)$, where $m(\alpha)=0$ and $m(t) := t^2+4\,t+5$. We have collected the points whose coordinates depend algebraically on all the conjugate roots of a same irreducible polynomial, $m(t)\in\mathbb{R}[t]$ (see the notion of conjugate points in [15]). Thus, from $\widetilde{\mathscr{P}}_{3,1}$, we obtain two rational parametrizations given by:
  - $\widetilde{\mathscr{P}}_{3,1a}(t) := (t^4, -2+i)$ which can be properly reparametrized as $\widetilde{\mathscr{M}}_{3,1}(t) := (t, -2+i)$ (see [13]).
  - $\widetilde{\mathscr{P}}_{3,1b}(t) := (t^4, -2-i)$ which can be reparametrized as $\widetilde{\mathscr{M}}_{3,1}(t) := (t, -2-i)$ (see [13]).

  Note that $\widetilde{\mathscr{P}}_{3,1a}(t)$ and $\widetilde{\mathscr{P}}_{3,1b}(t)$ are the parametrizations of two complex asymptotes $\widetilde{\mathscr{C}}_{3,1a}$ and $\widetilde{\mathscr{C}}_{3,1b}$.
- $\widetilde{\mathscr{P}}_{3,2}(t) := \left(t^3, \dfrac{3175\ldots113\sqrt[3]{103^2}\sqrt[3]{(-1915\ldots880\,\alpha+7311\ldots759)^2}\,\alpha t}{1299560599332066084366131105651465209158301\ldots664} - \right.$
$-\dfrac{1254510\ldots2965\sqrt[3]{103^2}\sqrt[3]{(-1915307\ldots0880\,\alpha+7311410\ldots6759)^2}t}{21659343322201101406102185094191086819305030423822147094 4} -$
$-\dfrac{13690955225473280\sqrt[3]{103}\sqrt[3]{-1915307\ldots70880\,\alpha+7311410\ldots26759}\,\alpha\,t^2}{26403789750400376029519659801} +$
$+\dfrac{8520846961643513\sqrt[3]{103}\sqrt[3]{-1915307\ldots970880\alpha+7311410\ldots926759}\,t^2}{176025265002669173530131065 34} +$
$\left.+\alpha t^3 - \dfrac{19425159575\,\alpha}{15549038208} + \dfrac{12229619291}{2591506368}\right)$, where $m(\alpha)=0$ and $m(t) := 209\,t^2 - 268\,t + 492$. From $\widetilde{\mathscr{P}}_{3,2}$, we obtain two rational parametrizations that define two complex asymptotes, $\widetilde{\mathscr{C}}_{3,2a}$ and $\widetilde{\mathscr{C}}_{3,2b}$.

- $\widetilde{\mathscr{P}}_{3,3}(t) := \left(t^3, (1/8 - 1/8\sqrt{3}\,\mathrm{i})\, t - 1/24\right)$ has complex coefficients, but it is important to highlight that this parametrization defines a real asymptote $\widetilde{\mathscr{C}}_{3,3}$ implicitly defined by the polynomial

$$\widetilde{f}_{3,3}(x, y) := -13824\, y^3 - 1728\, y^2 - 216\, x - 72\, y - 1 \in \mathbb{R}[x, y].$$

  In order to compute the implicit polynomial, we use the resultant method presented in [15].

The curve $\mathscr{C}_4$ is the only one with the lowest degree and the smaller number of monomials. $\mathscr{C}_4$ has two infinity branches and the highest degree asymptote has degree 2 and the parametrizations are

- $\widetilde{\mathscr{P}}_{4,1}(t) := \left(t^2,\ 11/6\sqrt{2}\, t + 65/72\right)$.
- $\widetilde{\mathscr{P}}_{4,2}(t) := (t, -2/5t - 5/4)$.

Table 2 shows that the curve $\mathscr{C}_3$ requires the longest execution time and generates the highest overhead in the microprocessor and memory when Algorithm 3.1 is run.

It is important to note that we should analyze the results obtained for the input curves $\mathscr{C}_{12}$ and $\mathscr{C}_{13}$, with degrees 13 and 9, respectively. Algorithm 3.1 has presented the highest performance, and the obtained asymptotes have degree 2.

The curve $\mathscr{C}_{12}$ has six complex asymptotes and one real linear asymptote. More precisely, the parametrizations defining the asymptotes are

$$\widetilde{\mathscr{P}}_{12,1}(t) := \left(t^2,\ \frac{965084862034\,\alpha^2 + 1657570956128\,\alpha - 10958170857228}{7490349\,\alpha^7\sqrt{\alpha}}\, t + \right.$$
$$+ \frac{15127222542616\,\alpha^2 + 5513812351100\,\alpha + 62931415515000}{7490349\,\alpha^{10}\sqrt{\alpha}}\, t -$$
$$- \frac{326894725 + 516120472\alpha + 64717862\alpha^2 - 83621175\alpha^3 + 27982896\alpha^4}{2496783000} -$$
$$\left. - \frac{1273427\alpha^5}{416130500}\right),$$

Table 2 Hardware resources required by Algorithm 3.1 for the case of implicit curves

Id.	CPU time	Real time	Memory used
$\mathscr{C}_1$	26.17 ms	25.06 ms	1978.84 KiB
$\mathscr{C}_2$	66.13 ms	31.56 ms	1386.33 KiB
$\mathscr{C}_3$	362.72 ms	283.47 ms	10806.75 KiB
$\mathscr{C}_4$	14.86 ms	16.88 ms	810.67 KiB
$\mathscr{C}_5$	168.47 ms	124.77 ms	3758.74 KiB
$\mathscr{C}_6$	8.78 ms	8.09 ms	695.43 KiB
$\mathscr{C}_7$	27.70 ms	39.57 ms	1495.48 KiB
$\mathscr{C}_8$	76.90 ms	40.75 ms	1924.63 KiB
$\mathscr{C}_9$	86.23 ms	45.01 ms	1755.44 KiB
$\mathscr{C}_{10}$	75.12 ms	51.87 ms	1333.29 KiB
$\mathscr{C}_{11}$	119.21 ms	82.42 ms	3706.32 KiB
$\mathscr{C}_{12}$	8.65 ms	7.93 ms	782.95 KiB
$\mathscr{C}_{13}$	7.13 ms	6.88 ms	602.33 KiB
$\mathscr{C}_{14}$	76.67 ms	52.47 ms	1875.11 KiB

where $m(\alpha) = 0$ and $m(t) := 3\,t^6 + 24\,t^5 - 172\,t^3 + 318\,t^2 + +300\,t + 1250.$.

- $\widetilde{\mathscr{P}}_{12,2}(t) := (t, 1/2)$ that is a real parametrization of the asymptote $\widetilde{\mathscr{C}}_{12,2}$.

Finally, the curve $\mathscr{C}_{13}$ has one real linear asymptote and four complex asymptotes. The parametric equations of these asymptotes are

- $\widetilde{\mathscr{P}}_{13,1}(t) := (t, 1)$.
- $\widetilde{\mathscr{P}}_{13,2}(t) := \left(t^2, \dfrac{3905\ldots16\,\alpha^3 + 2835\ldots00\,\alpha^2 + 30317\ldots16\,\alpha + 7637\ldots12}{\alpha^9\sqrt{\alpha}}\,t + \right.$
$\left. + \dfrac{-112 + 8\alpha - 98\alpha^2 + 3\alpha^3}{384\alpha}\right)$,
where $m(\alpha) = 0$ and $m(t) := t^4 - 32\,t^3 - 16\,t^2 - 256\,t + 64$. It is easy to check that two of these asymptotes are complex and the other two are real (we may compute the implicit polynomial defined by $\widetilde{\mathscr{P}}_{13,2}$ by using for instance the resultant method presented in [15]).

4.1.2 Computation of asymptotes of parametric algebraic curves

This subsection presents the results obtained by executing the methods described in the Appendix B, with an accuracy equal to ten for the Puiseux series expansion, for the case of the parametric curves defined in the Appendix A defined according to Equation 7

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

Table 3 shows that, for the case of curves expressed in parametric form, Algorithm 3.2 presents a lower efficiency than Algorithm 3.1. In most cases, this algorithm produces a higher system overhead and requires more hardware resources. Thus, it can be seen that Algorithm 3.2 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.

Thus, the greatest hardware resource requirements correspond to the calculation of the asymptotes of the curves $\mathscr{C}_3$ and $\mathscr{C}_{11}$, where the algorithm has needed more than 24 hours of CPU usage and more than 100 GiB of memory.

Also, the execution of the algorithm with the curve $\mathscr{C}_{13}$ requires 17 seconds of real time, 23 seconds of microprocessor usage time, and 2.55 GiB of memory; which contrasts sharply with applying Algorithm 3.1 to the implicit curve $\mathscr{C}_{13}$, 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 3.2 for the calculation of the asymptotes of the curve $\mathscr{C}_4$ presents the least degree of overload, even compared to the application of Algorithm 3.1 to the case of the implicit curve $\mathscr{C}_4$. 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. In the next section, we compare both algorithms in depth.

5 Results and discussion

In this section, a comparative analysis of Algorithms 3.1 and 3.2 is carried out. For this purpose, some properties of the input curves are considered, and the results obtained when constructing the respective asymptotes (see Appendix A).

Table 3 Hardware resources used by Algorithm 3.2 for the case of parametric curves

Id.	CPU time	Real time	Memory used
$\mathscr{C}_1$	16.13 ms	20.38 ms	2014.46 KiB
$\mathscr{C}_2$	131.45 ms	117.78 ms	17305.41 KiB
$\mathscr{C}_3$	$> 10^8$ ms	$> 10^8$ ms	$> 10^8$ KiB
$\mathscr{C}_4$	3.98 ms	3.60 ms	276.03 KiB
$\mathscr{C}_5$	114.12 ms	63.74 ms	10630.93 KiB
$\mathscr{C}_6$	2.57 ms	2.69 ms	288.92 KiB
$\mathscr{C}_7$	25.10 ms	26.58 ms	3193.96 KiB
$\mathscr{C}_8$	102.26 ms	72.53 ms	8634.75 KiB
$\mathscr{C}_9$	47.83 ms	25.44 ms	2832.28 KiB
$\mathscr{C}_{10}$	18496.00 ms	16028.00 ms	1795206.41 KiB
$\mathscr{C}_{11}$	$> 10^8$ ms	$> 10^8$ ms	$> 10^8$ KiB
$\mathscr{C}_{12}$	21726.00 ms	18961.00 ms	2720673.58 KiB
$\mathscr{C}_{13}$	22877.00 ms	17054.00 ms	2667242.91 KiB
$\mathscr{C}_{14}$	24.00 ms	24.00 ms	3668.91 KiB

Table 4 allows us to analyze 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 monomials, $\mathscr{C}_4$, to the curve with the highest complexity, $\mathscr{C}_3$. So, the best result for each algorithm is shown in bold.

Afterward, we illustrate in some figures the behavior of each algorithm from the values obtained in Table 4. The graphics illustrate the time, in milliseconds, and memory, in KiB, needed to execute Algorithm 3.1 (blue) and/or Algorithm 3.2 (pink) considering the input and also the output parameters of the algebraic curves defined implicitly and parametrically.

Figure 6 shows that analyzing the case of Algorithm 3.1, the highest efficiency is observed since it requires the least amount of CPU time. Likewise, it is shown that, for both algorithms, the degree of the curve determines the microprocessor time needed to build the asymptotes.

On the other hand, it is remarkable that although Algorithm 3.2 behaves well with simple curves (see curve $\mathscr{C}_6$ in Table 4), however, CPU usage time increases exponentially as the degree of the input curve increases (see curves $\mathscr{C}_3$ and $\mathscr{C}_{11}$ in Table 4).

The following graphics represent the behavior of each algorithm depending on the parameter of the input curve (axis x), concerning the CPU usage time expressed in milliseconds or memory capacity given in KiB, respectively (axis y). Note that this is a linear-logarithmic plot, based on 10, in which the trend lines are exponential.

Figure 7 shows the behavior of the algorithms considering the number of monomials (axis x). In this case, we can see that the number of monomials only influences Algorithm 3.1, however, it does not seem to have a significant impact when Algorithm 3.2 is run. In order to explain this point, we consider two aspects. The first one is that a curve parametrically defined always can be expressed implicitly and in this case, the number of monomials defining the implicit expression is, in general, higher. On the other side, the computation of the Puiseux series has more overload when the number of monomials increases (see the implementations of Appendix B, specifically the execution of the command `algcurves:-puiseux`) and also with the presence of conjugated roots. This point explains Fig. 10, which shows the

Table 4 Properties and hardware resources for implicit and parametric curves

Id	Deg	#Monom.		Asympt.				Hw. resources for implicit			Hw. resources for parametric		
		Imp	Par	#	MxDeg	#real	#complex	CPU_t	$Real_t$	Mem	CPU_t	$Real_t$	Mem
$\mathscr{C}_1$	7	21	5	5	2	3	2	26,17	25,06	1978,84	16,13	20,38	2014,46
$\mathscr{C}_2$	7	21	3	3	3	3	–	66,13	31,56	1386,33	131,45	117,78	17305,41
$\mathscr{C}_3$	17	114	5	5	3	1	4	362,72	283,47	10806,75	$> 10^8$	$> 10^8$	$> 10^8$
$\mathscr{C}_4$	3	8	3	2	2	2	–	14,86	16,88	810,67	3,98	3,60	**276,03**
$\mathscr{C}_5$	6	25	5	3	4	3	–	168,47	124,77	3758,74	114,12	63,74	10630,93
$\mathscr{C}_6$	4	11	3	3	2	3	–	8,78	8,09	695,43	**2,57**	**2,69**	288,92
$\mathscr{C}_7$	4	12	3	2	3	2	–	27,70	39,57	1425,48	25,10	26,58	3193,96
$\mathscr{C}_8$	5	13	3	2	2	2	–	76,90	40,75	1924,63	102,26	72,53	8634,75
$\mathscr{C}_9$	5	15	3	2	3	2	–	86,23	45,01	1755,44	47,83	25,44	2832,28
$\mathscr{C}_{10}$	7	19	3	3	3	1	2	75,12	51,87	1333,29	18496,00	16028,00	1795206,41
$\mathscr{C}_{11}$	15	58	4	4	5	2	2	119,21	82,42	3706,32	$> 10^8$	$> 10^8$	$> 10^8$
$\mathscr{C}_{12}$	13	31	3	7	2	1	6	8,65	7,93	782,95	21726,00	18961,00	2720673,58
$\mathscr{C}_{13}$	9	22	2	5	2	3	2	**7,13**	**6,88**	**602,33**	22877,00	17054,00	2667242,91
$\mathscr{C}_{14}$	5	15	2	2	2	2	–	76,67	52,47	1875,11	24,00	24,00	3668,91

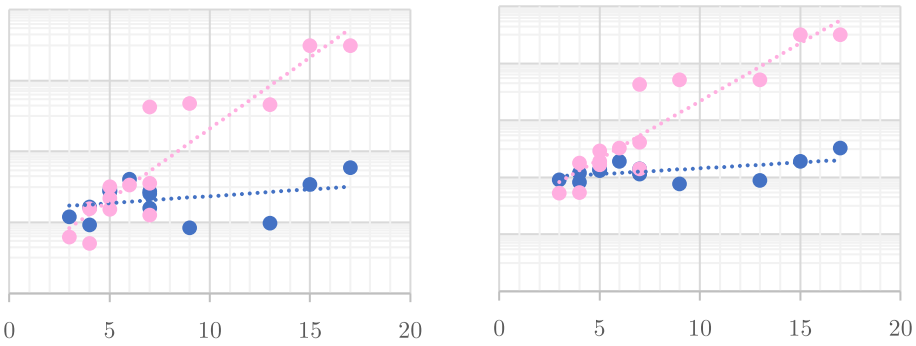

Fig. 6 CPU_t (left) *versus* memory (right) resources needed to run Algorithm 3.1 (blue) and Algorithm 3.2 (pink) depending on the curve degree.

machine requirements to construct the asymptotes of the input curves, depending on the number of the real or complex asymptotes (axis x). As we can see, the number of complex asymptotes clearly influences the performance of Algorithm 3.2, because executing the command

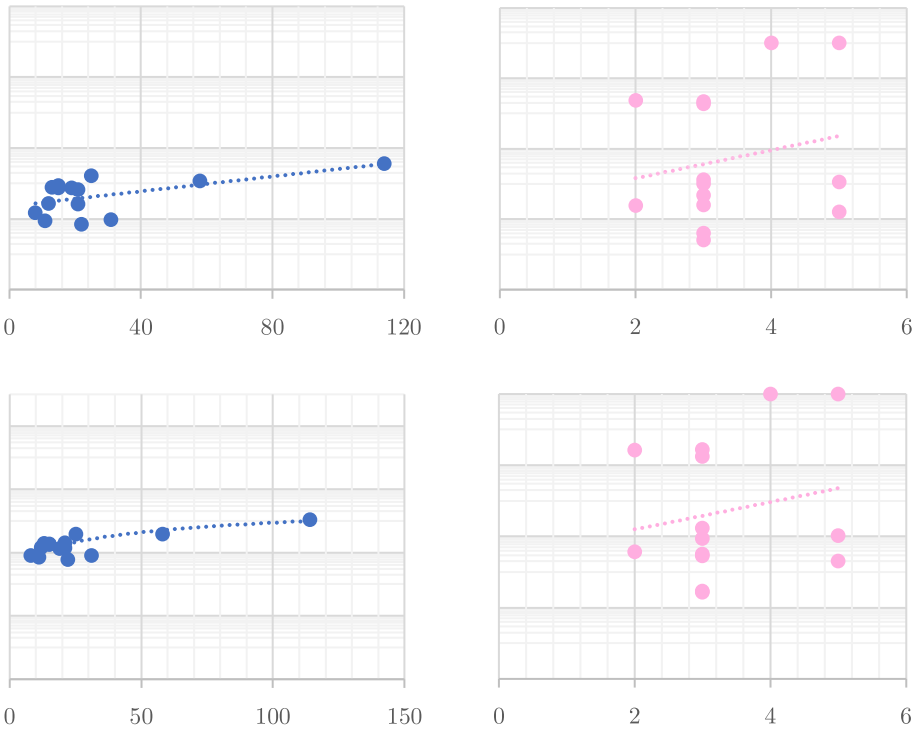

Fig. 7 CPU_t (top) *versus* memory (bottom) resources needed to run Algorithm 3.1 (blue) and Algorithm 3.2 (pink) depending on the number of monomials.

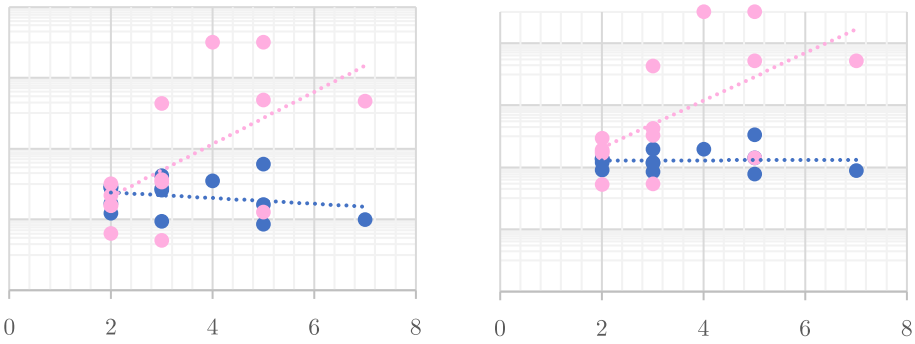

Fig. 8 CPU_t (left) *versus* memory (right) resources needed to run Algorithm 3.1 (blue) and Algorithm 3.2 (pink) depending on the number of branches.

`algcurves:-puiseux` on such input parameters can generate conjugate roots in some cases.

On the other hand, Fig. 8 shows the result for the number of branches (axis x). In this case, it does not seem that the number of the branches determines the time of the microprocessor or memory capacity for Algorithm 3.1, although it does influence the case of Algorithm 3.2. Note that number of branches is the same that the number of Puiseux solutions and also, the overload introduced by the command `algcurves:-puiseux` is higher for the parametric case.

Figure 9 shows that the system performance depends on the highest asymptotes degree. Both algorithms require more CPU time and memory capacity as the degree increases (remember that trend lines are exponential).

Finally, we can state that Algorithm 3.1 presents the best computational performance, requiring the least amount of hardware resources: CPU time, real execution time, and capacity of memory. Likewise, it has been shown that for both algorithms, the degree of the curve is the parameter that determines the overload and amount of system resources needed to build the asymptotes.

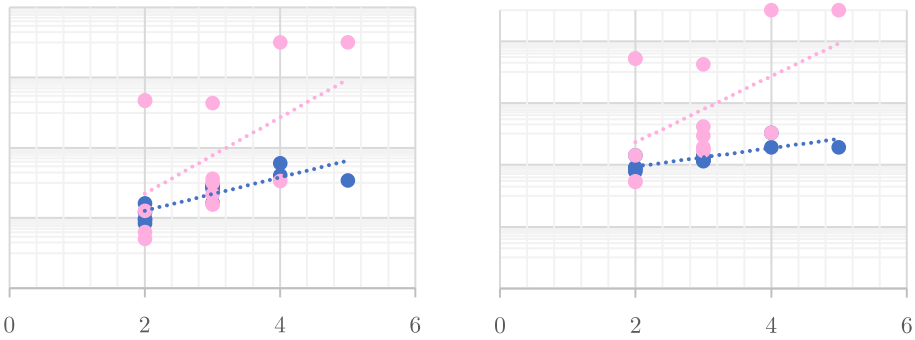

Fig. 9 CPU_t (left) *versus* memory (right) resources needed to run Algorithm 3.1 (blue) and Algorithm 3.2 (pink) depending on the highest asymptotes degree.

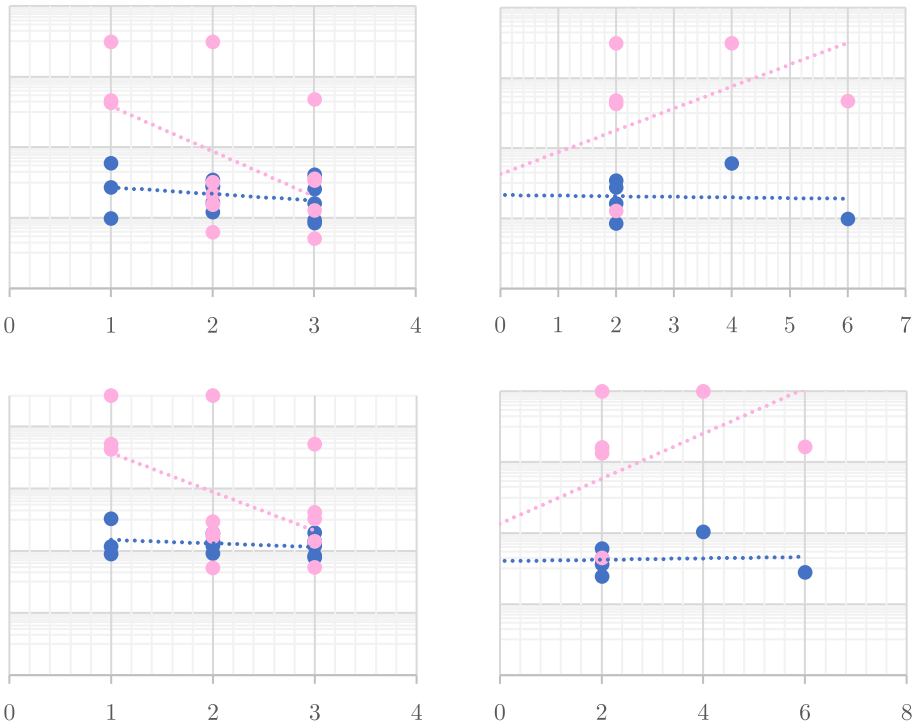

Fig. 10 CPU_t (top) *versus* memory (bottom) resources needed to run Algorithm 3.1 (blue) and Algorithm 3.2 (pink) depending on the number of the real or complex asymptotes.

It is note worthy, that Algorithm 3.2 has a good behavior with simple curves (see values for the curves $\mathscr{C}_4$ and $\mathscr{C}_6$ in Table 4). However, it is a very "heavy" algorithm when the degree of the input curve increases, in which case the hardware needs to grow up exponentially (see the curves $\mathscr{C}_3$ and $\mathscr{C}_{11}$ in Table 4).

Also, we have observed that the highest overhead occurs in the system when the parameter accuracy is increased (equal to $3-5-7-10$), although this improves accuracy, it also increases the overhead introduced by computing the Puiseux series.

Appendix A: Definition of parametric curves

- $\mathscr{P}_1(s) := \left(\dfrac{\mathrm{s}^2+1}{\left(\mathrm{s}^3-\mathrm{s}+1\right)\left(\mathrm{s}^2-1\right)^2}, \dfrac{2\mathrm{s}^3+5\mathrm{s}^2+1}{\mathrm{s}^5-2\mathrm{s}^3+\mathrm{s}^2+\mathrm{s}-1} \right).$
- $\mathscr{P}_2(s) := \left(\dfrac{\left(10\mathrm{s}^3-1\right)s^2}{\left(s^2-1\right)^2(s-2)^3}, \dfrac{s^3+s^2+3}{(\mathrm{s}-2)^2\left(s^2-1\right)} \right).$
- $\mathscr{P}_3(s) := \left(\dfrac{\mathrm{s}^8+2\mathrm{s}^4-\mathrm{s}^2-\mathrm{s}-1}{\left(\mathrm{s}^2+1\right)^4\left(\mathrm{s}^2+2\right)^3\mathrm{s}^3}, \dfrac{\mathrm{s}^7+\mathrm{s}^6-\mathrm{s}^5+\mathrm{s}^2+1}{\left(\mathrm{s}^2+2\right)^3\mathrm{s}} \right).$

- $\mathscr{P}_4(s) := \left(\dfrac{5+\mathrm{s}^2}{\mathrm{s}\,(\mathrm{s}-2)^2}, \dfrac{\mathrm{s}^2+3\mathrm{s}+1}{\mathrm{s}\,(\mathrm{s}-2)} \right)$.
- $\mathscr{P}_5(s) := \left(\dfrac{\mathrm{s}^4-\mathrm{s}^3+5\mathrm{s}^2+2\mathrm{s}+1}{\mathrm{s}^4\,(\mathrm{s}-1)\,(\mathrm{s}-2)}, \dfrac{2\mathrm{s}^4-3\mathrm{s}^3-2\mathrm{s}^2-26\mathrm{s}-18}{\mathrm{s}^4\,(\mathrm{s}-1)\,(\mathrm{s}-2)} \right)$.
- $\mathscr{P}_6(s) := \left(\dfrac{\mathrm{s}^3+2\mathrm{s}-1}{\left(\mathrm{s}^2-1\right)(\mathrm{s}-2)^2}, \dfrac{2\mathrm{s}^3+\mathrm{s}^2+1}{(\mathrm{s}-2)\left(\mathrm{s}^2-1\right)} \right)$.
- $\mathscr{P}_7(s) := \left(\dfrac{\mathrm{s}^3+2\mathrm{s}-1}{(\mathrm{s}-1)\,(\mathrm{s}-2)^3}, \dfrac{2\mathrm{s}^3+\mathrm{s}^2+1}{(\mathrm{s}-2)^2\,(\mathrm{s}-1)} \right)$.
- $\mathscr{P}_8(s) := \left(\dfrac{4\left(\mathrm{s}^2+1\right)}{\mathrm{s}^2\,(\mathrm{s}-2)^3}, \dfrac{\left(2\mathrm{s}^2+2\mathrm{s}-1\right)}{\mathrm{s}^2\,(\mathrm{s}-2)^2} \right)$.
- $\mathscr{P}_9(s) := \left(\dfrac{2\mathrm{s}^2-9}{s^2\,(\mathrm{s}+3)^3}, \dfrac{s^2+s-1}{s^2} \right)$.
- $\mathscr{P}_{10}(s) := \left(\dfrac{\mathrm{s}^3+2\mathrm{s}-1}{\left(\mathrm{s}^2+1\right)^2(\mathrm{s}-2)^3}, \dfrac{2\mathrm{s}^3+\mathrm{s}^2+1}{(\mathrm{s}-2)^2\left(\mathrm{s}^2+1\right)} \right)$.
- $\mathscr{P}_{11}(s) := \left(\dfrac{\mathrm{s}^5+2\mathrm{s}^2+\mathrm{s}-1}{\mathrm{s}^2\,(\mathrm{s}-1)^3\left(\mathrm{s}^2+1\right)^5}, \dfrac{2\mathrm{s}^4-\mathrm{s}^3+\mathrm{s}^2+1}{\left(\mathrm{s}^2+1\right)^3(\mathrm{s}-1)^2} \right)$.
- $\mathscr{P}_{12}(s) := \left(\dfrac{\mathrm{s}-1}{s\left(\mathrm{s}^6+2\right)^2}, \dfrac{\mathrm{s}^3-\mathrm{s}+1}{\left(\mathrm{s}^6+2\right)} \right)$.
- $\mathscr{P}_{13}(s) := \left(\dfrac{\mathrm{s}-1}{(s+1)\left(\mathrm{s}^4-2\right)^2}, \dfrac{\mathrm{s}^3}{\mathrm{s}^4-2} \right)$.
- $\mathscr{P}_{14}(s) := \left(\dfrac{\mathrm{s}^2+10}{s^2\,(s-2)^3}, \dfrac{s^2+1}{\mathrm{s}^2} \right)$.

Appendix B: Implementations

The following page shows the procedures which implement Algorithms 3.1 and 3.2 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·y^3 - 88 y^4 + 2352 x^2·y + 1368 x·y^2 + 360 y^3 - 1568 x^2 - 1960 x·y - 520 y^2 + 896 x
       + 368 y - 128 :
  AsymptotesImplicit(f, 3);
```

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

Procedure AsymptotesParametric (parametric_curve, accPuiseux)

DescriptionComputes 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 } *accPuiseux*
@return [[{expression}, {expression}], ...] parametric_asymptotes list

Code

```
> AsymptotesParametric := proc(parametric_curve, accPuiseux)
    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, p2/p1, 1/p1];                          # Projective curve
    g := [p2/p1, 1/p1 - t];
    ec_aux := numer(g[2]);
    ℓ := convert(puiseux(ec_aux, t=0, s, accPuiseux), set);
    for i to nops(ℓ) do
        aux_r := subs(s=ℓ[i], p2);
        aux_r := convert(series(aux_r, t=0, accPuiseux), polynom);
        r := subs(t=z^-1, aux_r);
        e_index := EIndex(r);
        r := expand(simplify(subs(z=z^e_index, r), radical, symbolic));
        tilde_r := NonNegativeOps(r);
        parametric_asymptoteslist := [t^e_index, subs(z=t, 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 := AsymptotesParametric(P, 4);
```

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

Acknowledgements 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)and by the European Union, Digital Europe Program 21-22 Call Cloud Data and TEF (CitCom.ai 101100728). The author S. Pérez-Díaz belongs to the Research Group ASYNACS (Ref. CCEE2011/R34).

Author Contributions The authors contributed equally to this work.

Funding Ministerio de Ciencia, Innovación y Universidades - Agencia Estatal de Investigación/PID2020-113192GB-I00 (Mathematical Visualization: Foundations, Algorithms and Applications)
State Plan for Scientific and Technical Research and Innovation of the Spanish MCI (PID2021-127946OB-I00) and by the European Union, Digital Europe Program 21-22 Call Cloud Data and TEF (CitCom.ai 101100728).

Data Availability All data generated or analysed during this study are included in this published article.

Declarations

Conflicts of interest The authors declare that they have no conflict of interest.

References

1. Blasco, A., Pérez-Díaz, S.: Asymptotes and perfect curves. Computer Aided Geometric Design **31**(2), 81–96 (2014). https://doi.org/10.1016/j.cagd.2013.12.004
2. Blasco, A., Pérez-Díaz, S.: Asymptotic behavior of an implicit algebraic plane curve. Computer Aided Geometric Design **31**(7), 345–357 (2014). https://doi.org/10.1016/j.cagd.2014.04.002
3. Blasco, A., Pérez-Díaz, S.: Asymptotes of space curves. Journal of Computational and Applied Mathematics **278**, 231–247 (2015). https://doi.org/10.1016/j.cam.2014.10.013
4. Blasco, A., Pérez-Díaz, S.: A new approach for computing the asymptotes of a parametric curve. Journal of Computational and Applied Mathematics **364**, 112350–118 (2020). https://doi.org/10.1016/j.cam.2019.112350
5. Bradley, E., Stolle, R.: Automatic construction of accurate models of physical systems. Annals of Mathematics and Artificial Intelligence **17**, 1–28 (1996). https://doi.org/10.1007/BF02284622
6. Fernández de Sevilla, M., Magdalena Benedicto, R., Pérez-Díaz, S.: Computing branches and asymptotes of meromorphic functions.Revista de la Real Academia de Ciencias Exactas, Físicas y Naturales, Serie A Matemáiticas. **117**(109) (2023). https://doi.org/10.1007/s13398-023-01441-7
7. 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. Computer Aided Geometric Design 94, 102084 (2022). https://doi.org/10.1016/j.cagd.2022.102084
8. Cheever, E.A., Li, Y.: A tool for construction of bode diagrams from piecewise linear asymptotic approximations. International Journal of Engineering Education **21**(2), 335–340 (2005)
9. Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: A tutorial. Computer **29**(3), 31–44 (1996). https://doi.org/10.1109/2.485891
10. Kečkić, J.D.: A method for obtaining asymptotes of some curves. The Theaching of Mathematics **III**(1), 53–59 (2000)
11. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–444 (2015). https://doi.org/10.1038/nature14539
12. Maxwell, E.A.: An Analytical Calculus, vol. 3. Cambridge University Press, Cambridge, United Kingdom (1962)
13. Pérez-Díaz, S.: On the problem of proper reparametrization for rational curves and surfaces. Computer Aided Geometric Design **23**(4), 307–323 (2006). https://doi.org/10.1016/j.cagd.2006.01.001
14. Salas, S.L., Etgen, G.J., Hilles, E.: Calculus. One and Several Variables. John Wiley and Sons Inc, New Jersey, NJ, USA (1995)
15. Sendra, J.R., Winkler, F., Pérez-Díaz, S.: Rational Algebraic Curves: A Computer Algebra Approach. Algorithms and Computation in Mathematics, Springer, Berlin, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73725-4
16. Zeng, G.: Computing the asymptotes for a real plane algebraic curve. Journal of Algebra **316**(2), 680–705 (2007). https://doi.org/10.1016/j.jalgebra.2007.03.030