mathematics

MDPI

Article

The Newton–Puiseux Algorithm and Triple Points for Plane Curves

Stefano Canino [1], Alessandro Gimigliano [2,*] and Monica Idà [2]

[1] Dipartimento di Scienze Matematiche, Politecnico di Torino, 10129 Torino, Italy; stefano.canino@polito.it
[2] Dipartimento di Matematica, Università di Bologna, 40126 Bologna, Italy; monica.ida@unibo.it
[*] Correspondence: alessandr.gimigliano@unibo.it

Abstract: The paper is an introduction to the use of the classical Newton–Puiseux procedure, oriented towards an algorithmic description of it. This procedure allows to obtain polynomial approximations for parameterizations of branches of an algebraic plane curve at a singular point. We look for an approach that can be easily grasped and is almost self-contained. We illustrate the use of the algorithm first in a completely worked out example of a curve with a point of multiplicity 6, and secondly, in the study of triple points on reduced plane curves.

Keywords: plane curves singularities; Newton–Puiseux algorithm; triple points

MSC: 14H20; 14B05; 13J05

Citation: Canino, S.; Gimigliano, A.; Idà, M. The Newton–Puiseux Algorithm and Triple Points for Plane Curves. *Mathematics* **2023**, *11*, 2324. https://doi.org/10.3390/math11102324

Academic Editors: Elena Guardo, Emeritus Mario Gionfriddo, Salvatore Milici, Martin Schlichenmaier and Adolfo Ballester-Bolinches

Received: 18 January 2023
Revised: 2 April 2023
Accepted: 5 May 2023
Published: 16 May 2023

1. Introduction

The study of singular points of algebraic plane curves is a very classical subject in algebraic geometry, and a renewed interest in this subject has come from its connections with applications (e.g., in CAD). Rational curves have been of particular interest from this point of view, since they can be parameterized and so they are particularly suited for plotting a design. Hence, finding ways to describe plane rational curves at their singular points (more specifically, finding algorithms that can do that) has been quite a prominent subject of research; many results have been obtained by attacking this problem from different points of view (e.g., see [1–3]).

When the curve is not rational, one cannot rely on the advantage of a parameterized curve, nevertheless, there are local parameterizations of the branches of the curve at one of its points, albeit not given by polynomials, but rather by power series. Namely, if the curve is defined by a polynomial $f(x,y) \in \mathbb{C}[x,y]$, with $f(0,0) = 0$, one can find parameterizations $T \to (T^m, p(T))$, $p(T) \in \mathbb{C}\{T\}$, the ring of convergent power series in T, such that $f(T^m, p(T)) = 0$, $p(0) = 0$. Every such parameterization (where $p(T)$ gives a holomorphic function defined in a neighborhood of the point) describes a branch of $f(x,y)$ at $(0,0)$. For example, the curve $\mathcal{C}$, defined by $2xy^2 - y^3 + x^2y^2 - 2x^6 + x^5y + x^7 - x^6y + x^8 - x^9 = 0$, has two branches at $O = (0,0)$, given by $(T, 2T + \dots)$ and $(T^2, T^5 + \dots)$. The first branch is a smooth one, while the second has a double point (a ramphoid cusp).

The way to obtain these parameterizations is via the Puiseux series and the Newton–Puiseux algorithm; these are quite a classical subject of study (the idea goes back to Newton, then was rediscovered by Puiseux, see [4–6]), but their knowledge is not so widespread. Hence, we consider it worthwhile to present a concise and self-contained introduction to their use which could be useful, especially for graduate students and young researchers.

We work on $\mathbb{C}$ and we consider a Puiseux series as a power series where positive rational exponents also appear; we denote by $\mathbb{C}\{\{x\}\}$ the ring of such series (for this ring there is not a standard symbol). In some texts, the name Puiseux series is used for series where also a finite number of negative rational exponents is allowed and in this case, Puiseux series would form a field K (namely, the quotient field of our ring $\mathbb{C}\{\{x\}\}$). With this approach, the core result of this theory can be stated as *The field K is algebraically closed*. This theorem, also called the Newton–Puiseux Theorem, can be found with a concise but not constructive proof in [7] and with a different and somewhat simpler (but again non-constructive) proof in [8]. From this point of view, the branches of an algebraic curve given by $f \in \mathbb{C}[x,y]$ are given by the linear factors of f as an element of $K[y]$: $f = a\prod_{i=1}^{m}(y-\overline{y}_i)$, where a, and the $\overline{y}_i$'s (not necessarily distinct) are in K, while m is the multiplicity of $\mathcal{C}$ at O (e.g., see the exposition in [9]).

Detailed treatises of the theory (apart from the early references of Newton and Puiseux themselves) can be found for example in ([chs. 6, 7], [10]), ([chs. 7, 8 and appx. 4], [11]), ([ch. 1 sec. 3], [12]), ([ch. IV],[9]). A constructive approach, in order to find explicitly the parameterization of the branches, is given by the *Newton–Puiseux Algorithm*; e.g., this is the way the Newton–Puiseux Theorem is proved in [9] (but also there the algorithm is not completely constructive, see ([Intro], [13])).

For a more constructive approach, which also considers non-algebrically closed fields or finite ones instead of just $\mathbb{C}$, one can consider [13–16], and the references therein, where also discussions about the search for efficient computational versions of the algorithm can be found.

The first aim of this paper is to expound a handy and compact version of the Newton–Puiseux procedure (over $\mathbb{C}$), which avoids involving negative exponents and is given by an algorithm "as concise as possible", together with a self-contained justification of why it actually works (which avoids the proof of the Newton–Puiseux Theorem, although we use [9] as the main reference for some details) and a "Stop criterion" which allows to obtain good polynomial approximations of the series which give the local branches of the curve.

The plan of the paper is as follows: after a section of preliminaries (Section 2), in Section 3, we describe our version of the Newton–Puiseux Algorithm for the branches of a reduced algebraic curve at a singular point. The way we proceed is the following: first, we describe two essential tools for the Algorithm: the $*$*-procedure* and a graph G_f associated to the curve. Such a graph G_f will guide the steps of the Algorithm. The next essential tool for the Algorithm is the "Stop criterion", which we describe immediately after, and we show how it works in making G_f a finite graph (so that the procedures can come to an end). After that, we have all the main ingredients for the Algorithm and we have also pointed out what is still missing to obtain a complete justification of why it works; in the following Section 3.3, we present all the necessary lemmata which answer those remaining questions (this subsection is quite technical, so a reader who is less interested in such matter can skip it and still have a good understanding of how the Algorithm works). Eventually, in Section 3.4, we write the formal Algorithm (Algorithm 1) and then an easier version of it (Algorithm 2), which can be used in case one already knows the irreducible decomposition of the curve, also when f is not reduced. In order to fully illustrate the way the given Algorithm works, in Section 3.5, we present an example of its application on a curve which possesses several branches at a point of multiplicity 6.

In Section 4, we present an application of Algorithm 1, which is the second aim of the paper: in Theorem 4.3, we use it in order to obtain the description of all possible structures of a triple point with a triple tangent (of course, one can apply the Algorithm also if the triple point has two or three tangents, but in this case, the possibilities for its structure are well known). Moreover, when $\mathcal{C}$ is of cuspidal type, i.e., it has only a 3-branch, we see that its parameterization can be characterized by an invariant which we call its *type* (see Definition 8). The type is a number $s \in \mathbb{N}$ with $s \geq 4$ and $s \not\equiv 0 \pmod 3$ and the triple point is of type s if and only if the curve is analytically equivalent to $(T^3, a_sT^s + \dots)$ and topologically equivalent to (T^3, T^s).

2. Preliminaries

In the following, $\mathcal{C}$ is a reduced plane algebraic curve in $\mathbb{A}^2(\mathbb{C})$, of equation $f = 0$, where $f \in \mathbb{C}[x,y]$ (we write for short C: $f = 0$) and $O := (0,0)$ is a point of $\mathcal{C}$; we denote with $m_O(\mathcal{C})$ the multiplicity of $\mathcal{C}$ at O. Given a formal power series $s(T) \in \mathbb{C}[[T]]$, $o(s)$ denotes the order of $s(T)$.

Definition 1. *A formal parameterization of $\mathcal{C}$ at O is a pair $(p(T), q(T))$, with $p(T), q(T) \in \mathbb{C}[[T]]$, such that $f(p(T), q(T)) = 0, p(0) = 0$ and $q(0) = 0$. Two formal parameterizations, (p_1, q_1) and (p_2, q_2), are equivalent if there exists an invertible series $s \in \mathbb{C}[[T]]$ such that $(p_1, q_1) = (p_2 \circ s, q_2 \circ s)$. Moreover, a formal parameterization is called reduced if it is not equivalent to a formal parameterization of the form $(p(T^m), q(T^m)), m > 1$.*

Definition 2. *A branch of $\mathcal{C}$ at O is a class of equivalent formal reduced parameterizations of $\mathcal{C}$ at O.*

Remark 1. *It can be shown (see, for example, ([ch. IV, Theorem 2.2], [9])) that in a suitable coordinate system any given parameterization is equivalent to one of the type*

$$(T^n, p(T))$$

where $p(T) \in \mathbb{C}[[T]]$. Moreover, two such parameterizations

$$(T^n, p(T)), \quad (T^m, q(T))$$

are equivalent if and only if $n = m$ and there exists an n-th root of the unity ω such that $p(\omega T) = q(T)$.

Remark 2. *Sometimes, what we defined as a branch of $\mathcal{C}$ at O is called a place of $\mathcal{C}$ at O and it can be viewed as the algebraic counterpart of the more usual analytical definition of branch. However, the two definitions are equivalent; more details can be found in (pp. 96–97, [9]).*

Definition 3. *Let $R = (p(T), q(T))$ be a branch of $\mathcal{C}$ at O. We define the* multiplicity of R at *O to be the number $m = m_O(R) := \min\{o(p), o(q)\}$ and we say that R is an m-branch of $\mathcal{C}$. Moreover, if c and d are the coefficients of T^m, respectively, in p and q, the line passing through O with direction (c, d) is called the tangent line of R.*

Remark 3. *It can be checked that the equation of the tangent cone to $\mathcal{C}$ at O is the product of the tangent lines (possibly repeated) to the different branches of $\mathcal{C}$ at O, with each one raised to the multiplicity of the corresponding branch. It follows that:*

- *$\mathcal{C}$ has in each of its points a finite number of branches;*
- *If t is a tangent of $\mathcal{C}$ at O with multiplicity m_t and $R_1, \dots, R_s$ are the branches of $\mathcal{C}$ at O whose tangent is t, then $\sum_{i=1}^{s} m_O(R_i) = m_t$*
- *If $R_1, \dots, R_n$ are the branches of $\mathcal{C}$ at O, then $\sum_{i=1}^{n} m_O(R_i) = m_O(\mathcal{C})$.*

In order to find explicitly the branches of a curve, we will use the Newton–Puiseux Algorithm, which uses the Newton polygon and Puiseux series, which are nothing but power series where positive rational exponents are admitted. More formally, we have

Definition 4. *A Puiseux series is an element $p \in \mathbb{C}\{\{x\}\} := \bigcup_{r=1}^{\infty} \mathbb{C}[[x^{\frac{1}{r}}]]$. The minimum r such that $p \in \mathbb{C}[[x^{\frac{1}{r}}]]$ is called the lowest common denominator of $p \in \mathbb{C}\{\{x\}\}$ (it is also called order of polydromy, e.g., see [17]). The order of p is defined analogously to the formal series order.*

Notice that a finite sum of Puiseux series is a Puiseux series, since if $s_1 \in \mathbb{C}[[x^{\frac{1}{r_1}}]], \dots,$ $s_n \in \mathbb{C}[[x^{\frac{1}{r_n}}]]$, then $s_1 + \cdots + s_n \in \mathbb{C}[[x^{\frac{1}{r_1 \cdots r_n}}]]$.

Definition 5. *A Puiseux y-polynomial is a polynomial $f \in \mathbb{C}\{\{x\}\}[y]$ of the form*

$$f(x,y) = \sum_{i=0}^{n} p_i(x)y^i$$

where each $p_i(x)$ has a finite number of terms, i.e., $p_1 \in \mathbb{C}[x^{\frac{1}{r_1}}], \ldots, p_n \in \mathbb{C}[x^{\frac{1}{r_n}}]$. Setting $r = r_1 \cdots r_n$, we have that $p_1, \ldots, p_n \in \mathbb{C}[x^{\frac{1}{r}}]$, hence, f is polynomial in $\mathbb{C}[x^{\frac{1}{r}}, y]$.

A root of f is a series $p(x^{\frac{1}{q}})$ such that $f(x, p(x^{\frac{1}{q}})) = 0$.

Definition 6. *A Puiseux y-polynomial $f \in \mathbb{C}[x^{\frac{1}{r}}, y]$ is called Newton convenient if $f(O) = 0$ and both $x^{\frac{1}{r}}$ and y do not divide f (i.e., there are $a, b \in \mathbb{N}$ such that the monomials $x^{\frac{a}{r}}$ and y^b appear in f with non-zero coefficients).*

We want to define the Newton polygon associated to such an f; the definition can actually be given for polynomials in any number of variables and, in general, is called *Newton diagram* (e.g., see ([Definition 2.14 pp. 121–122],[12])). We use only the two-dimensional Newton diagram for Newton-convenient Puiseux y-polynomials and we refer to it as *Newton polygon* (see also [9]).

Definition 7. *Let $f = \sum_{i+j=1}^{d} a_{ij}(x^{\frac{1}{r}})^i y^j \in \mathbb{C}[x^{\frac{1}{r}}, y]$ be a Newton-convenient Puiseux y-polynomial and let $supp(f) := \{(\frac{i}{r}, j) \in \mathbb{Q}^2, a_{ij} \neq 0\}$ be the support of f. The* Newton Polygon *of f, denoted by $\Gamma(f)$, is defined as follows:*

- *Let P_1 be the leftmost point of $supp(f)$ (notice that P_1 is on the y-axis; if there are more points with abscissa 0, we choose the lowest one). From P_1, we rotate a vertical downward ray counterclockwise and stop rotating it when it meets the first point of $supp(f)$. We call P_2 the rightmost point of $supp(f)$ met by this ray;*
- *From P_2, we rotate a vertical downward ray counterclockwise and we stop rotating it when it meets the first point of $supp(f)$. We call P_3 the rightmost point of $supp(f)$ met by this ray;*
- *We repeat this procedure until we reach a point P_{k+1} on the x-axis.*

For all P_n, P_{n+1}, we denote by $\boldsymbol{b}_n$ the segment $\overline{P_n P_{n+1}}$; we set $\Gamma(f) = \{\boldsymbol{b}_1 \ldots \boldsymbol{b}_k\}$ and we say that $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k$ are the edges of $\Gamma(f)$. The difference between the ordinates of P_n and P_{n+1} is called height *of $\boldsymbol{b}_n$. For each $\boldsymbol{b} \in \Gamma(f)$, we define the $\boldsymbol{b}$-truncation of f as*

$$f^{\boldsymbol{b}} = \sum_{(\frac{i}{r}, j) \in \boldsymbol{b} \cap \mathbb{Q}^2} a_{ij}(x^{\frac{1}{r}})^i y^j.$$

Remark 4. *We would like to point out a few easy properties of the Newton polygon which will be useful in the following: in the notation of the previous definition, consider the unbounded subset N of the first quadrant, $x \geq 0, y \geq 0$, whose border is given by the Newton Polygon, the ray from P_1 on the y-axis and the ray from P_{k+1} on the x-axis. Then, by construction, N is convex. Hence, it is clear that there are neither vertical nor horizontal edges in the Newton Polygon and that all the edges have negative slope.*

Example 1. Let us consider the following polynomial $f \in \mathbb{C}[x^{\frac{1}{2}}, y]$:

$$f = 2y^6 + x^{\frac{3}{2}}y^4 - 3x^3y^3 + x^{\frac{7}{2}}y^4 + 3x^4y^3 - x^{\frac{9}{2}}y^2 + x^{\frac{11}{2}}y^2 - 4x^7y^2 + x^{\frac{15}{2}}y + x^{\frac{19}{2}}y - 7x^{12} + 4x^{\frac{25}{2}}.$$

Its Newton Polygon is given in Figure 1:

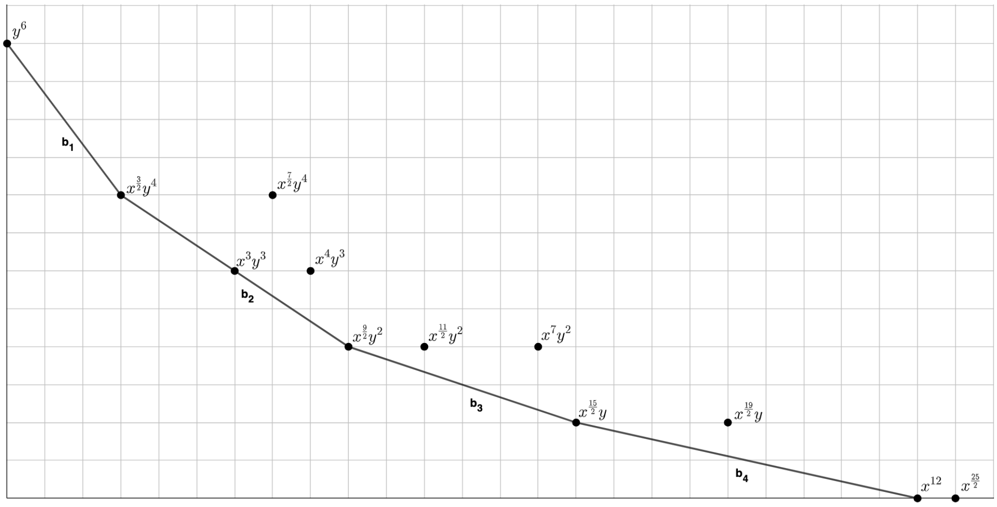

Figure 1. The Newton Polygon of f.

and we write $\Gamma(f) = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4\}$. Moreover, we have:

$$f^{\boldsymbol{b}_1} = 2y^6 + x^{\frac{3}{2}}y^4, \quad f^{\boldsymbol{b}_2} = x^{\frac{3}{2}}y^4 - 3x^3y^3 - x^{\frac{9}{2}}y^2, \quad f^{\boldsymbol{b}_3} = -x^{\frac{9}{2}}y^2 + x^{\frac{15}{2}}y, \quad f^{\boldsymbol{b}_4} = x^{\frac{15}{2}}y - 7x^{12.}$$

The next theorem is well known and it will be very useful in the following (its proof can be found, for example, in ([Theorem 6.4], [10])).

Theorem 1 (Implicit Function Theorem for Puiseux series)**.** *Let*

$$f = \sum_{i=0}^{d} p_i(x)y^i \in \mathbb{C}\{\{x\}\}[y].$$

If $p_0(0) = 0$ and $p_1(0) \neq 0$, then $\exists!\ p \in \mathbb{C}\{\{x\}\}$ such that $p(0) = 0$ and $f(x, p(x)) = 0$. Moreover, if $p_i \in \mathbb{C}[[x^{\frac{1}{r}}]]$ for $i = 1, \dots, d$, then $p \in \mathbb{C}[[x^{\frac{1}{r}}]]$.

3. The Newton–Puiseux Algorithm

The aim of this section is to present our version of the Newton–Puiseux Algorithm: for a reduced polynomial $f = \sum_{i+j=1}^{d} a_{ij}x^iy^j$, it yields an approximation of all the branches at O of the algebraic plane curve $\mathcal{C}$: $f = 0$. With this in mind, we present some preliminaries needed for the algorithm.

3.1. The $*$-Procedure and the Graph $\mathbf{G}_f$

In the following, h denotes a Puiseux y-polynomial with $h(O) = 0$, $h \in \mathbb{C}[x^{\frac{1}{r}}, y]$, and such that $x^{\frac{1}{r}} \nmid h$ (recall that $r \in \mathbb{N}, r > 0$). Hence, we consider powers of x with rational exponents, while the powers of y always have an integer exponent.

For the moment, we are just going to describe this procedure, which is the main tool for our Algorithm; later on (Section 3.3), we shall give the needed justifications in detail. In the procedure, we distinguish two cases, according to whether $y \mid h$ or not.

Case $\mathbf{y} \nmid \mathbf{h}$: By assumption, $x^{\frac{1}{r}} \nmid h$, $h(O) = 0$ and $y \nmid h$, hence, h is Newton-convenient. We consider the Newton Polygon $\Gamma(h) = \{\mathbf{b}_1, \dots, \mathbf{b}_k\}$ of h. We set

$$h^{(i)} := \frac{h^{\mathbf{b}_i}}{x^{u_i}y^{v_i}}; \qquad \forall\, i = 1, \dots, k$$

where x^{u_i} and y^{v_i} are respectively the highest power of x and y such that $h^{(i)} \in \mathbb{C}[x^{\frac{1}{r}}, y]$. Moreover, we have $h^{(i)}(O) = 0$ (this will be proved later).

We find all the s_i roots $q_{i1}, \dots q_{is_i}$ of $h^{(i)}$, considered as a polynomial in y; if $-\frac{1}{r_i}$ is the slope of $\mathbf{b}_i$, then $h^{(i)}$ has all its roots in $\mathbb{C}[x^{r_i}]$ and each q_{ij} is of the form $q_{ij}(x) = c_{ij}x^{r_i}$ with $c_{ij} \in \mathbb{C}$ (this will be proved later).

We finally set

$$h^{(ij)} := \frac{h(x, x^{r_i}(c_{ij} + z))}{x^{m_i}}, \qquad \forall\, i = 1, \dots, k, \quad \forall\, j = 1, \dots, s_i$$

where m_i is the greatest power of x such that $h^{(ij)} \in \mathbb{C}\{\{x\}\}[z]$.

By construction, $h^{(ij)}$ is a Puiseux z-polynomial, and if $h^{(ij)}$ is in, say, $\mathbb{C}[x^{\frac{1}{b}}, y]$, then $x^{\frac{1}{b}} \nmid h$. Moreover, we have $h^{(ij)}(O) = 0$ (this will be proved later). Therefore, the result of the procedure is:

$$s_i = s_i(h), \quad q_{ij} = q_{ij}(h), \quad c_{ij} = c_{ij}(h), \quad r_i = r_i(h), \quad h^{(ij)}, \quad i = 1, \dots, k, \quad j = 1, \dots, s_i.$$

<u>Case $\mathbf{y} \mid \mathbf{h}$</u>: Let y^e be the greatest power of y such that $\hat{h} := \frac{h(x,y)}{y^e} \in \mathbb{C}\{\{x\}\}[y]$. We have two subcases: $\hat{h}(O) \neq 0$ and $\hat{h}(O) = 0$.

- If $\hat{h}(O) \neq 0$, we ignore $\hat{h}$ and we take into account uniquely the contribution given by the root $y = 0$ of h, in the following way: we define a virtual Newton Polygon $\Gamma(h)$ with just one edge and we set, with an abuse of notation:

$$\Gamma(h) := \{\mathbf{b}_1\}, \quad s_1 := 1, \quad q_{11} := 0, \quad c_{11} = 0, \quad r_1 := 0, \quad h^{(11)} := 0.$$

- If $\hat{h}(O) = 0$, we take into account the contribution given by $\hat{h}$ as well as the one given by the root $y = 0$, in the following way: $\hat{h}$ being Newton-convenient, we can consider the Newton Polygon $\Gamma(\hat{h}) = \{\mathbf{b}_1 \dots \mathbf{b}_k\}$ of $\hat{h}$ and apply the procedure defined in Case $y \nmid h$ to the Puiseux y-polynomial $\hat{h}$; we rename the results of the procedure as follows:

$$s_i(h) := s_i(\hat{h}),\ q_{ij}(h) = q_{ij}(\hat{h}),\ c_{ij}(h) := c_{ij}(\hat{h}),\ r_i(h) := r_i(\hat{h}),\ h^{(ij)} := \hat{h}^{(ij)},\ i = 1, \dots, k,\ j = 1, \dots, s_i.$$

Moreover, we add a further virtual edge $\mathbf{b}_{k+1}$ and we set, with an abuse of notation:

$$\Gamma(h) := \{\mathbf{b}_1, \dots, \mathbf{b}_k, \mathbf{b}_{k+1}\}, \quad s_{k+1} := 1, \quad q_{(k+1)1} := 0, \quad c_{(k+1)1} = 0, \quad r_{k+1} := 0, \quad h^{((k+1)1)} := 0.$$

Summarizing, the input of the $*$-procedure is a Puiseux y-polynomial $h \in \mathbb{C}[x^{\frac{1}{r}}, y]$ such that $x^{\frac{1}{r}} \nmid h$ and $h(O) = 0$, and the output is made of:

$$s_i = s_i(h) \in \mathbb{N}, \quad r_i = r_i(h) \in \mathbb{Q}_+, \quad c_{ij} = c_{ij}(h) \in \mathbb{C}$$

and some Puiseux z-polynomials $h^{(ij)} \in \mathbb{C}\{\{x\}\}[z]$, for any choice of the edge $\mathbf{b}_i$ of $\Gamma(h)$ and for any q_{ij} relative to that edge.

Moreover, each polynomial $h^{(ij)}$ in the output either satisfies the necessary conditions to apply again the $*$-procedure, or $h^{(ij)} = 0$.

Example 2. Let us apply the $*$-procedure to the polynomial

$$h = y^2(2y^6 + x^{\frac{3}{2}}y^4 - 3x^3y^3 + x^{\frac{7}{2}}y^4 + 3x^4y^3 - x^{\frac{9}{2}}y^2 + x^{\frac{11}{2}}y^2 - 4x^7y^2 + x^{\frac{15}{2}}y + x^{\frac{19}{2}}y - 7x^{12} + 4x^{\frac{25}{2}}) \in \mathbb{C}[x^{\frac{1}{2}}, y].$$

Following the notation of the $*$-procedure, we are in case $y|h$ with $e = 2$ and

$$\hat{h} = 2y^6 + x^{\frac{3}{2}}y^4 - 3x^3y^3 + x^{\frac{7}{2}}y^4 + 3x^4y^3 - x^{\frac{9}{2}}y^2 + x^{\frac{11}{2}}y^2 - 4x^7y^2 + x^{\frac{15}{2}}y + x^{\frac{19}{2}}y - 7x^{12} + 4x^{\frac{25}{2}}$$

so $\hat{h}(O) = 0$ and we apply the $*$-procedure to $\hat{h}$. By Example 1, we know the Newton Polygon of $\hat{h}$ and $\Gamma(\hat{h}) = \{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3, \mathbf{b}_4\}$, with

$$\hat{h}^{\mathbf{b}_1} = 2y^6 + x^{\frac{3}{2}}y^4, \quad \hat{h}^{\mathbf{b}_2} = x^{\frac{3}{2}}y^4 - 3x^3y^3 - x^{\frac{9}{2}}y^2, \quad \hat{h}^{\mathbf{b}_3} = -x^{\frac{9}{2}}y^2 + x^{\frac{15}{2}}y, \quad \hat{h}^{\mathbf{b}_4} = x^{\frac{15}{2}}y - 7x^{12}.$$

As a consequence, we have

$$(u_1, v_1) = (0,4), \quad (u_2, v_2) = \left(\frac{3}{2}, 2\right), \quad (u_3, v_3) = \left(\frac{9}{2}, 1\right), \quad (u_4, v_4) = \left(\frac{15}{2}, 0\right)$$

and thus

$$\hat{h}^{(1)} = \frac{2y^6 + x^{\frac{3}{2}}y^4}{y^4} = 2y^2 + x^{\frac{3}{2}}, \quad \hat{h}^{(2)} = \frac{x^{\frac{3}{2}}y^4 - 3x^3y^3 - x^{\frac{9}{2}}y^2}{x^{\frac{3}{2}}y^2} = y^2 - 3x^{\frac{3}{2}}y - x^3$$

$$\hat{h}^{(3)} = \frac{-x^{\frac{9}{2}}y^2 + x^{\frac{15}{2}}y}{x^{\frac{9}{2}}y} = -y + x^3, \quad \hat{h}^{(4)} = \frac{x^{\frac{15}{2}}y - 7x^{12}}{x^{\frac{15}{2}}} = y - 7x^{\frac{9}{2}}.$$

By computing the roots of the $\hat{h}^{(i)}$'s, considered as polynomials in y, we find:

$$q_{11}(c\hat{h}) = \frac{\sqrt{2}}{2}ix^{\frac{3}{4}}, \quad q_{12}(\hat{h}) = -\frac{\sqrt{2}}{2}ix^{\frac{3}{4}}$$

$$q_{21}(\hat{h}) = \frac{3+\sqrt{13}}{2}x^{\frac{3}{2}}, \quad q_{22}(\hat{h}) = \frac{3-\sqrt{13}}{2}x^{\frac{3}{2}}$$

$$q_{31}(\hat{h}) = x^3, \quad q_{41}(\hat{h}) = 7x^{\frac{9}{2}}$$

and we can compute the $\hat{h}^{(ij)}$'s just applying the definition but, since they are very long to write, we avoid giving them explicitly. At the end of the procedure, we add the virtual edge $\mathbf{b}_5$ and we have our output:
the $s_i(h)$'s:

$$s_1(h) = 2, \quad s_2(h) = 2, \quad s_3(h) = 1, \quad s_4(h) = 1, \quad s_5(h) = 1$$

the $q_{ij}(h)$'s:

$$q_{11}(h) = \frac{\sqrt{2}}{2}ix^{\frac{3}{4}}, \quad q_{12}(h) = -\frac{\sqrt{2}}{2}ix^{\frac{3}{4}}, \quad q_{21}(h) = \frac{3+\sqrt{13}}{2}x^{\frac{3}{2}}, \quad q_{22}(h) = \frac{3-\sqrt{13}}{2}x^{\frac{3}{2}}$$

$$q_{31}(h) = x^3, \quad q_{41}(h) = 7x^{\frac{9}{2}}, \quad q_{51}(h) = 0$$

the $c_{ij}(h)$'s:

$$c_{11}(h) = \frac{\sqrt{2}}{2}i, \quad c_{12}(h) = -\frac{\sqrt{2}}{2}i, \quad c_{21}(h) = \frac{3+\sqrt{13}}{2}, \quad c_{22}(h) = \frac{3-\sqrt{13}}{2}$$

$$c_{31}(h) = 1, \quad c_{41}(h) = 7, \quad c_{51}(h) = 0$$

the $r_i(h)$'s:

$$r_1(h) = -\frac{3}{4}, \quad r_2(h) = -\frac{2}{3}, \quad r_3(h) = -\frac{1}{3}, \quad r_4(h) = -\frac{2}{9}, \quad r_5(h) = 0$$

and the $h^{(ij)}$'s, with $h^{(ij)} = \hat{h}^{(ij)}$ for $i = 1,2,3,4$ and $h^{(51)} = 0$.

Now, let $f \in \mathbb{C}[x,y]$ be a given polynomial, with $f(O) = 0$ and $x \nmid f$. We apply the $*$-procedure successively to all the polynomials $f, f^{(ij)}, f^{(ij)\,(lt)}, \ldots$ obtaining a graph $\mathbf{G}_f$, as described in Figure 2.

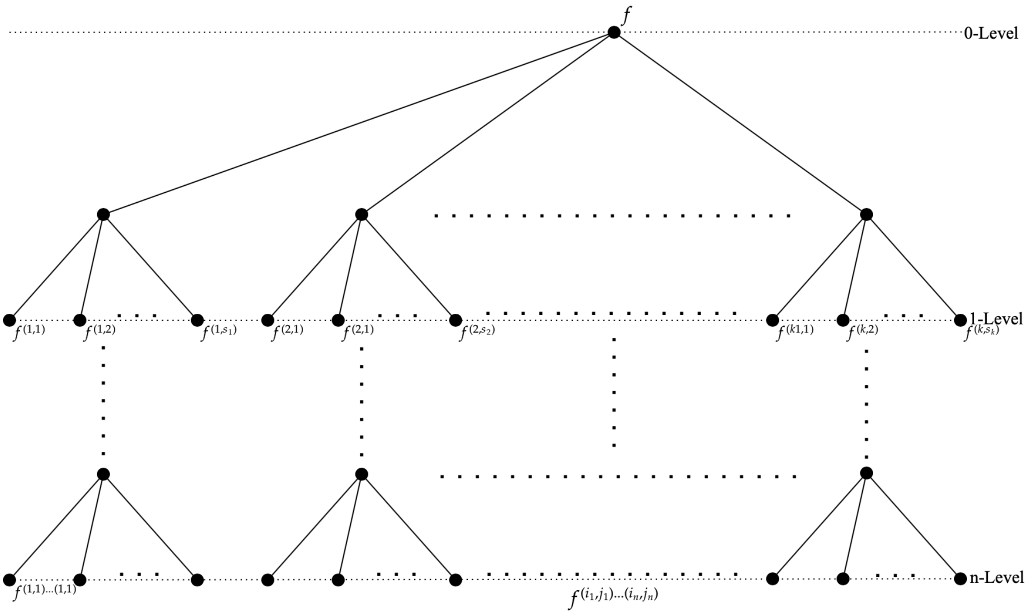

Figure 2. The graph $\mathbf{G}_f$.

The 0-level consists only of the vertex f, the 1-level consists of the vertices $f^{(ij)}$ and so on, so that the n-level is composed of the vertices marked with $2n$ indices.

Since, in general, Puiseux series have infinite terms, the $*$-procedure, in general, does not end without a stop criterion.

Stop criterion 3.1. *In order to have that $\mathbf{G}_f$ is a finite graph, we stop the procedure every time that we meet a polynomial $\overline{f}$ such that one of the following conditions occurs:*

(i) $\overline{f} = 0$*;*

(ii) $\overline{f}$ *satysfies the assumptions of Theorem 1, i.e., the monomial z appears in* $\overline{f}$ *with a non-zero coefficient (graphically, the Newton Polygon* $\Gamma(\overline{f})$ *has a unique edge of height one).*

3.2. The Paths on $\mathbf{G}_f$ and the Branches of the Curve

We proceed considering every possible descending path γ in $\mathbf{G}_f$, starting from $f_0 := f$ and stopping when we reach an $f^{(i_1\,j_1)\dots(i_n\,j_n)}$, which satisfies (i) or (ii) of Section 3.1 above. Notice that, in general, we obtain paths of different lengths.

For each path γ, we call *n-th step* the passage from the n-level to the $(n+1)$-level of γ; graphically, it is given by 3 vertices and 2 edges: the highest vertex is a polynomial $f^{(i_1 j_1)\dots(i_n j_n)}$ which from now on is denoted by $f_n(\gamma)$ and, since the $*$-procedure introduces a new variable at each step, we consider $f_n(\gamma) \in \mathbb{C}\{\{x\}\}[y_n]$. The middle vertex is associated with an edge of $\Gamma(f_n)$ which is now called $\mathbf{a}_n(\gamma)$ and the lowest one is now called $f_{n+1}(\gamma)$. When no confusion is possible, we will just write $f_n, \mathbf{a}_n, f_{n+1}$.

Once a path is fixed, we use the following notation for the $*$-procedure at the n-th step, assuming that $f_n \in \mathbb{C}[x^{\frac{1}{r}}, y_n]$, where r depends on γ and n, and that $\mathbf{a}_n$ is not a "virtual edge" which sprouts from a zero root:

- x^{u_n} and $y_n^{v_n}$ are, respectively, the highest power of x and y_n such that:

$$g_n = \frac{f_n^{\mathbf{a}_n}(x, y_n)}{x^{u_n} y_n^{v_n}} \in \mathbb{C}\{\{x\}\}[y_n];$$

- $q_n = c_n x^{r_n}$ is the chosen root of g_n;
- m_n is the highest power of x such that

$$f_{n+1} = \frac{f_n(x, x^{r_n}(c_n + y_{n+1}))}{x^{m_n}} \in \mathbb{C}\{\{x\}\}[y_{n+1}].$$

If, instead, $\mathbf{a}_n$ is a "virtual edge", we only have $c_n = 0,\ r_n = 0,\ f_{n+1} = 0$ (so the path γ stops here by the stop criterion).

3.2.1. Each Path γ Gives the Approximation of a Branch

Let us consider an algebraic curve $\mathcal{C}$, associated to $f(x, y) = 0$ and passing through the origin. In the notation above, each path γ gives the approximation of a branch of $\mathcal{C}$ at O in the following way:

If $\overline{n}$ is the lenght of the path γ, we set

$$p_\gamma(x) := \sum_{n=0}^{\overline{n}} c_n x^{r_0+\cdots+r_n}.$$

If no confusion arises we just write $p(x)$ for $p_\gamma(x)$.

Let r be the common lowest denominator of $p(x)$; Theorem 1 guarantees that $p(x)$ is the truncation of a Puiseux series $\tilde{p}(x)$, with common lowest denominator r. Moreover, the way the algorithm is constructed guarantees that $f(x, \tilde{p}(x)) = 0$ and $\tilde{p}(0) = 0$, so that $(T^r, p(T^r))$ is the approximation of a branch $(T^r, \tilde{p}(T^r))$ of $\mathcal{C}$ at O. For more details, see (pp. 98–99, [9]).

The above procedure is summarized in the forthcoming Algorithm 1. Notice that in the Algorithm we start with a reduced curve $\mathcal{C} : f(x, y) = 0$, i.e., the polynomial $f = f_0$ is supposed to be without multiple factors. This, the forthcoming Lemma 6 implies that the case relative to a "virtual edge" (which would occur when $f_{\overline{n}-1}$ has a zero root, forcing the path γ to stop) can never occur, unless $\mathcal{C}$ has a branch coming from a component that is a rational curve, i.e., there is an algebraic branch at O. In other words, if (i) of the Stop criterion 3.1 occurs, it means that $y_{\overline{n}-1} \mid f_{\overline{n}-1}$, i.e., $0 \in \mathbb{C}\{\{x\}\}$ is a root of $f_{\overline{n}-1}$: hence, we can stop following the path in question because the corresponding Puiseux series has a finite number of terms and we have found all of them.

Of course, the roots of f have, in general, infinite terms. Every time we apply the $*$-procedure, we obtain a new polynomial which has a Newton Polygon whose height is $\leq$the previous one by Lemma 3. The criterion actually yields to a stop on any descending path in $\mathbf{G}_f$ because f is reduced and this implies that each path on the graph ends with a step of height 1 (see also (p. 105, [9])), hence, we end in case (ii) of the step, unless we ended in case (i) before.

If, once arrived at the $\bar{n}^{th}$ step, we are in case (ii) of the Stop criterion 3.1, i.e., $f_{\overline{n}}(x, y_{\overline{n}})$ satisfies the hypothesis of Theorem 1, we have that for any $n \geq \overline{n}$, there is a unique choice for c_n and r_n. This means that the root $p(x)$ does not split anymore, i.e., if we were to go on along the path, from now on, there would be a unique possible choice at each step. Hence, we have found enough terms of $p_\gamma(x)$ to distinguish it from the other branches, therefore, we can stop the algorithm. Clearly, if one wants to calculate more terms of $p(x)$, it is enough to continue the procedure.

In conclusion, the Stop criterion 3.1 is a good one.

3.2.2. Justification of the Procedure

There are several results that are needed in order to show that we can actually perform the $*$-procedure described in Section 3.1, namely:

(1) In the $*$-procedure, we need $h^{(i)}(O) = 0$, and the roots $q_{i1}, \ldots q_{is_i}$ of $h^{(i)}$ need to be in $\mathbb{C}[x^{\frac{1}{r}}]$, with q_{ij} of the form $q_{ij}(x) = c_{ij}x^{r_i}$, $c_{ij} \in \mathbb{C}$;

(2) In order to keep applying the $*$-procedure, we must show that the Puiseux z-polynomial $h^{(ij)}$ satisfies $h^{(ij)}(O) = 0$.

Rephrasing (1) and (2) in the language used above in this Section 3.2, we must show that:

(1) $g_n(O) = 0$, and the roots $q_{i1}, \ldots q_{is_i}$ of g_n need to be in $\mathbb{C}[x^{\frac{1}{r}}]$, with q_{ij} of the form $q_{ij}(x) = c_{ij}x^{r_i}$, $c_{ij} \in \mathbb{C}$;

(2) $f_{n+1}(O) = 0$.

Finally, notice that we need to distinguish when the parameterizations we obtain in the algorithm are equivalent, in which case they give the same branch. Lemma 1 provides a criterion to do that. The fact that (1) and (2) above hold will be proved by Lemma 3 and Lemma 2 in the next section, respectively.

3.3. The Lemmata

This subsection is of a more technical sort. We present the lemmata which justify why the algorithm actually works.

Lemma 1. *Let $\mathcal{C} : f(x,y) = 0$ be a reduced curve such that $x \nmid f$ and let*

$$p(x^{\frac{1}{r}}) = \sum_{i=0}^{\infty} c_i (x^{\frac{1}{r}})^i, \qquad p'(x^{\frac{1}{r'}}) = \sum_{i=0}^{\infty} c'_i (x^{\frac{1}{r'}})^i$$

be two roots of f. Moreover, let γ and γ' be the paths associated to p and p' with the respective notation and set $\overline{n} = \min\{n \in \mathbb{N} \mid f_n(x, y_n)$ and $f'_n(x, y_n)$ satisfy Theorem 1$\}$. Then the parameterizations

$$(T^r, p(T)), \qquad (T^{r'}, p'(T))$$

are equivalent if and only if $r = r'$ and it exists ω such that $\omega^r = 1$ and

$$\sum_{i=0}^{\overline{n}} c_i (\omega x^{\frac{1}{r}})^i = \sum_{i=0}^{\overline{n}} c'_i (x^{\frac{1}{r}})^i.$$

Proof. If the parameterizations are equivalent, we conclude by Remark 1. By ([Ch. IV, Theorem 4.1], [9]), it follows that since $p(x^{\frac{1}{r}})$ is a root of $f(x,y)$, then also $p(\omega x^{\frac{1}{r}})$ is a root of $f(x,y)$, where $\omega^r = 1$. Moreover, since $f'_{\overline{n}}$ satisfies the assumptions of Theorem 1, there exists a unique root $q(x^{\frac{1}{r}})$ of $f(x,y)$ such that

$$q(x^{\frac{1}{r}}) = \sum_{i=0}^{\overline{n}} c'_i (x^{\frac{1}{r}})^i + \dots$$

hence, this root has to be $p'(x^{\frac{1}{r}})$. By assumption, we have

$$p(\omega x^{\frac{1}{r}}) = \sum_{i=0}^{\infty} c_i (\omega x^{\frac{1}{r}})^i = \sum_{i=0}^{\overline{n}} c_i (\omega x^{\frac{1}{r}})^i + \dots = \sum_{i=0}^{\overline{n}} c'_i (x^{\frac{1}{r}})^i + \dots$$

so that $p(\omega x^{\frac{1}{r}}) = q(x^{\frac{1}{r}})$. Hence, we have $p(\omega x^{\frac{1}{r}}) = p'(x^{\frac{1}{r}})$ and the result follows by Remark 1. □

Lemma 2. *Notation as in Section 3.2. Let $\mathbf{a}_n \in \Gamma(f_n)$, let ρ denotes its slope, and let $c_1 \dots c_t$ be the distinct roots of $g_n(1, y_n)$, of multiplicity $m_1, \dots, m_t$. Then, $g_n(O) = 0$, and the roots of $g_n(x, y_n) \in \mathbb{C}\{\{x\}\}[y_n]$ are $q_{n1}(x) = c_1 x^{r_n}, \dots, q_{nt}(x) = c_t x^{r_n}$, where $r_n = -\frac{1}{\rho}$; more precisely, there is a constant α such that*

$$g_n(x, y_n) = \alpha \prod_{s=1}^{t} (y_n - c_s x^{r_n})^{m_s}$$

where $\sum_{i=1}^{t} m_i$ is equal to the height of $\mathbf{a}_n$. We say that m_s is the multiplicity of c_s as a root of g_n.

Proof. We can assume that $y \nmid f_n$, hence f_n is Newton-convenient and we can consider its Newton Polygon $\Gamma(f_n)$; moreover, all its edges have strictly negative slopes by Remark 4.

Hence, if $(i_1, j_1), \ldots, (i_r, j_r)$ are the points of $\mathbf{a}_n$ ordered from left to right, we have $i_1 < i_2 < \cdots < i_r$ and $j_1 > j_2 > \cdots > j_r$. Therefore, if we set

$$f_n(x, y_n) = \sum_{(i,j) \in \mathrm{supp}(f_n)} a_{ij} x^i y_n^j$$

we have

$$f_n^{\mathbf{a}_n} = \sum_{k=1}^{r} a_{i_k j_k} x^{i_k} y_n^{j_k}, \qquad u_i = i_1, v_i = j_s$$

which in turn gives

$$g_n(x, y_n) = \sum_{k=1}^{r} a_{i_k j_k} x^{i_k - i_1} y_n^{j_k - j_r}$$

so we can see that $g_n(O) = 0$. The slope of $\mathbf{a}_n$ is $\rho = \frac{j_r - j_1}{i_r - i_1}$; we consider the possible roots of $g_n(x, y_n) \in \mathbb{C}\{\{x\}\}[y_n]$ of the form $y_n = ux^{r_n}$, $u \in \mathbb{C}$, $r_n = -\frac{1}{\rho}$. We have

$$g_n(x, y_n) = g_n(x, ux^{r_n}) = \sum_{k=1}^{r} a_{i_k j_k} x^{i_k - i_1 + r_n(j_k - j_r)} u^{j_k - j_r}.$$

Since $(i_k, j_k) \in \mathbf{a}_n,\ k = 1, \ldots, r$, the following equation holds:

$$(j_1 - j_r)(i_k - i_1) = (j_k - j_1)(i_1 - i_r), \qquad k = 1, \ldots, r$$

and therefore,

$$i_k - i_1 + r_n(j_k - j_r) = \frac{(i_k - i_1)(j_r - j_1) - (j_k - j_r)(i_r - i_1)}{j_r - j_1} = i_r - i_1, \qquad k = 1, \ldots, r.$$

Hence, we have

$$g_n(x, y_n) = x^{i_r - i_1} \sum_{k=1}^{r} a_{i_k j_k} u^{j_k - j_r}.$$

Now, we define

$$h(y_n) := g_n(1, y_n) = \sum_{k=1}^{r} a_{i_k j_k} y_n^{j_k - j_r}.$$

Since $h(y_n) \in \mathbb{C}[y_n]$ and $\deg(h) = j_1 - j_r$, $h(y_n)$ factorizes as

$$h(y_n) = a_{i_1 j_1} \prod_{s=1}^{t} (y_n - c_s)^{m_s}, \qquad with c_i \neq c_j \text{ and } \sum_{s=1}^{t} m_s = j_1 - j_r.$$

Thus, we have

$$g_n(x, y_n) = x^{i_r - i_1} h(u) = a_{i_1 j_1} x^{i_r - i_1} \prod_{s=1}^{t} (u - c_s)^{m_s} = a_{i_1 j_1} x^{i_r - i_1} \prod_{s=1}^{t} (\frac{y_n}{x^{r_n}} - c_s)^{m_s}.$$

Since

$$r_n \sum_{s=1}^{t} m_s = r_n(j_1 - j_r) = i_r - i_1,$$

we obtain

$$g_n(x, y_n) = a_{i_1 j_1} \prod_{s=1}^{t} x^{r_n m_s} (\frac{y_n}{x^{r_n}} - c_s)^{m_s} = a_{i_1 j_1} \prod_{s=1}^{t} (y_n - c_s x^{r_n})^{m_s}.$$

Therefore, the roots of g_n are

$$q_{n1} = c_1 x^{r_n},\ \dots,\ q_{nt} = c_t x^{r_n}$$

and we conclude the proof by observing that $j_1 - j_r = \sum_{s=1}^{t} m_s$ is exactly the height of $\mathbf{a}_n$. $\square$

Lemma 3. *Notation as in Section 3.2. If cx^{r_n} is a root of $g_n(x, y_n)$ of multiplicity m, then the lowest pure power of y_{n+1} in $f_{n+1}(x, y_{n+1})$ is y_{n+1}^m. Moreover, $f_{n+1}(O) = 0$.*

Proof. We use the same notation introduced in the proof of Lemma 2. We can write f_n as

$$f_n = f_n^{\mathbf{a}_n} + \underbrace{(f_n - f_n^{\mathbf{a}_n})}_{\tilde{f}_n} = x^{i_1} y_n^{j_r} g_n + \tilde{f}_n.$$

Hence, it follows that

$$f_n(x, x^{r_n}(c + y_{n+1})) = \underbrace{x^{i_1 + r_n j_r}(c + y_{n+1})^{j_r} g_n(x, x^{r_n}(c + y_{n+1}))}_{A} + \tilde{f}_n(x, x^{r_n}(c + y_{n+1})).$$

By Lemma 2

$$g_n(x, x^{r_n}(c + y_{n+1})) = a_{i_1 j_1} \prod_{s=1}^{t} (x^{r_n} y_{n+1} + (c - c_s) x^{r_n})^{m_s} = a_{i_1 j_1} x^{r_n \sum_{s=1}^{t} m_s} \prod_{s=1}^{t} (c - c_s + y_{n+1})^{m_s}$$

and

$$r_n = \frac{i_1 - i_r}{j_r - j_1} \quad \text{and} \quad \sum_{s=1}^{t} m_s = j_1 - j_r.$$

Without loss of generality, we can suppose that $c = c_1$ (and, as a consequence, that $m = m_1$). Therefore, we have

$$g_n(x, x^{r_n}(c + y_{n+1})) = a_{i_1 j_1} x^{i_r - i_1} y_{n+1}^{m} \prod_{s=2}^{t} (c - c_s + y_{n+1})^{m_s}$$

and, by substituting this in A, we obtain

$$A = a_{i_1 j_1} x^{r_n j_r + i_r} y_{n+1}^{m} (c + y_{n+1})^{j_r} \prod_{s=2}^{t} (c - c_s + y_{n+1})^{m_s}.$$

Now, let $a_{\bar{i}\bar{j}} x^{\bar{i}} y_n^{\bar{j}}$ be a monomial of $\tilde{f}_n$. Its evaluation in $(x, x^{r_n}(c + y_{n+1}))$ is

$$a_{\bar{i}\bar{j}} x^{\bar{i} + r_n \bar{j}} (c + y_{n+1})^{\bar{j}}.$$

Since $(\bar{i}, \bar{j}) \in (\operatorname{supp}(f_n) \setminus \mathbf{a}_n)$, the point $(\bar{i}, \bar{j})$ does not belong to $\mathbf{a}_n$. Let $\tilde{a}_n$ be the line containing the edge $\mathbf{a}_n$; $\tilde{a}_n$ has equation $i + r_n j = i_r + r_n j_r$. Since $\Gamma(f_n)$ is part of the border of the convex hull of $\operatorname{supp}(f_n)$, the point $(\bar{i}, \bar{j})$ is in the half plane $i + r_n j > i_r + r_n j_r$, hence,

$$\bar{i} + r_n \bar{j} > i_r + r_n j_r. \tag{1}$$

This shows that the power of x for which we have to divide $f_n(x, x^{r_n}(c + y_{n+1}))$ is $x^{r_n j_r + i_r}$ and that the pure powers of y_{n+1} in f_{n+1} are uniquely the ones coming from

$$\frac{A}{x^{r_n j_r + i_r}} = a_{i_1 j_1} y_{n+1}^{m} (c + y_{n+1})^{j_r} \prod_{s=2}^{t} (c - c_s + y_{n+1})^{m_s}.$$

Hence, the monomial with the lowest pure power of y_{n+1} is

$$\left(a_{i_1 j_1} c^{j_r} \prod_{s=2}^{t} (c - c_s)^{m_s}\right) y_{n+1}^{m}$$

so the lowest pure power of y_{n+1} in $f_{n+1}(x, y_{n+1})$ is y_{n+1}^m. We have

$$f_{n+1}(x, y_{n+1}) = \frac{A}{x^{r_n j_r + i_r}} + \frac{\tilde{f}_n(x, x^{r_n}(c + y_{n+1}))}{x^{r_n j_r + i_r}}$$

and $\frac{A}{x^{r_n j_r + i_r}}(O) = 0$ since it is divided by y_{n+1}, while $\frac{\tilde{f}_n(x, x^{r_n}(c+y_{n+1}))}{x^{r_n j_r + i_r}}(O) = 0$ by (1). □

Lemma 4. *Notation as in Section 3.2. Let $n \in \mathbb{N}$ be the minimum such that there exists $t \in \mathbb{N}$, $t \geq 1$ such that $y_{n+1}^t \mid f_{n+1}$. Then:*

$$y_{n+1}^t \mid f_{n+1} \ in\ \mathbb{C}\{\{x\}\}[y_{n+1}] \Leftrightarrow (y - \sum_{i=0}^{n} c_i x^{r_i + r_{i-1} + \cdots + r_0})^t \mid f \ in\ \mathbb{C}\{\{x\}\}[y]$$

Proof. Let

$$f_n(x, y_n) = \sum_{(i,j) \in \mathrm{supp}(f_n)} a_{ij} x^i y_n^j \quad ; \qquad f_{n+1}(x, y_{n+1}) = \sum_{(i',j') \in \mathrm{supp}(f_{n+1})} b_{i'j'} x^{i'} y_{n+1}^{j'}$$

and recall that the exponents of x are rationals, while those of y are integers.
$\Rightarrow$) Since $y_{n+1}^t \mid f_{n+1}(x, y_{n+1})$ it exists $h_{n+1} \in \mathbb{C}\{\{x\}\}[y_{n+1}]$ such that $f_{n+1}(x, y_{n+1}) = y_{n+1}^t h_{n+1}(x, y_{n+1})$. Moreover, we have

$$f_n(x, y_n) = x^{m_n} f_{n+1}\Big(x, \frac{y_n}{x^{r_n}} - c_n\Big) = x^{m_n} \Big(\frac{y_n}{x^{r_n}} - c_n\Big)^t h_{n+1}\Big(x, \frac{y_n}{x^{r_n}} - c_n\Big) =$$
$$= (y_n - c_n x^{r_n})^t \underbrace{x^{m_n - t r_n} h_{n+1}\Big(x, \frac{y_n}{x^{r_n}} - c_n\Big)}_{h_n(x, y_n)}.$$

Now, we want to show that $h_n(x, y) \in \mathbb{C}\{\{x\}\}[y_n]$. We have

$$h_{n+1}(x, y_{n+1}) = \frac{f_{n+1}(x, y_{n+1})}{y_{n+1}^t} = \sum_{(i',j') \in \mathrm{supp}(f_{n+1})} b_{i'j'} x^{i'} y_{n+1}^{j'-t}$$

therefore,

$$h_n(x, y_n) = x^{m_n - t r_n} h_{n+1}\Big(x, \frac{y_n}{x^{r_n}} - c_n\Big) = x^{m_n - t r_n} \sum_{(i',j') \in \mathrm{supp}(f_{n+1})} b_{i'j'} x^{i'} \Big(\frac{y_n}{x^{r_n}} - c_n\Big)^{j'-t} =$$
$$= \sum_{(i',j') \in \mathrm{supp}(f_{n+1})} b_{i'j'} x^{m_n - t r_n + i' - r_n(j'-t)} (y_n - c_n x^{r_n})^{j'-t}.$$

Since $j' - t \geq 0$, it is enough to prove that

$$m_n - t r_n + i' - r_n(j' - t) \geq 0, \quad \forall \ (i', j') \in \mathrm{supp}(f_{n+1}). \tag{2}$$

By definition

$$f_{n+1}(x, y_{n+1}) = \frac{f_n(x, x^{r_n}(c_n + y_{n+1}))}{x^{m_n}} = \sum_{(i,j) \in \mathrm{supp}(f_n)} a_{ij} x^{i + j r_n - m_n} (c_n + y_{n+1})^j =$$

$$= \sum_{(i,j)\in \operatorname{supp}(f_n)} a_{ij} x^{i+jr_n-m_n} \sum_{k=0}^{j} \binom{j}{k} y_{n+1}^{k} c_n^{j-k}$$

hence

$$(i',j') \in \operatorname{supp}(f_{n+1}) \Rightarrow \exists \ (i,j) \in \operatorname{supp}(f_n), k \in \mathbb{N}, k \leq j \text{ such that } i' = i + jr_n - m_n \text{ and } j' = k.$$

Thus, in order that (2) holds, it suffices that

$$m_n - tr_n + i + jr_n - m_n - r_n(k-t) = i + (j-k)r_n \geq 0 \quad \forall \ (i,j) \in \operatorname{supp}(f_n), 0 \leq k \leq j$$

and this inequality holds because $i, (j-k), r_n \geq 0$. Thus, we showed that $h_n(x, y_n) \in \mathbb{C}\{\{x\}\}[y]$.

Now, by the same reasoning, we have

$$f_{n-1}(x,y_{n-1}) = x^{m_{n-1}} f_n\left(x, \frac{y_{n-1}}{x^{r_{n-1}}} - c_{n-1}\right) = x^{m_{n-1}} \left(\frac{y_{n-1}}{x^{r_{n-1}}} - c_{n-1} - c_n x^{r_n}\right)^t h_n\left(x, \frac{y_{n-1}}{x^{r_{n-1}}} - c_{n-1}\right) =$$

$$= (y_{n-1} - c_{n-1}x^{r_{n-1}} - c_n^{r_n+r_{n-1}})^t \underbrace{x^{m_{n-1}-tr_{n-1}} h_n\left(x, \frac{y_{n-1}}{x^{r_{n-1}}} - c_{n-1}\right)}_{h_{n-1}(x,y_{n-1}) \in \mathbb{C}\{\{x\}\}[y_{n-1}]}.$$

Hence, since $f = f_0$, iterating the same procedure, we find $h(x,y) \in \mathbb{C}\{\{x\}\}[y]$ such that

$$f(x,y) = \left(y - \sum_{i=0}^{n} c_i x^{r_i + r_{i-1} + \cdots + r_0}\right)^t h(x,y).$$

$\Leftarrow$) The proof is analogous to the previous one. □

Lemma 5. *Let us consider $f, g \in \mathbb{C}[u,v]$ such that g is irreducible, $\deg g > 0$, and let $t \in \mathbb{N}, t > 0$. Then, $g^t \mid f$ if and only if g divides f and all its partial derivatives up to the order of $t-1$.*

Proof. $\Rightarrow$) Is trivial.
$\Leftarrow$) Let us prove the statement for $t = 1$. Let $f = gh$; since $g \mid f_u$ and $g \mid f_v$, there exist $h_1, h_2 \in \mathbb{C}[u,v]$ such that $f_u = gh_1$ and $f_v = gh_2$. We have

$$gh_1 = f_u = (gh)_u = g_u h + g h_u, \qquad gh_2 = f_v = (gh)_v = g_v h + g h_v.$$

Thus, we have that $g(h_1 - h_u) = g_u h$ and $g(h_2 - h_v) = g_v h$ so that $g \mid g_u h$ and $g \mid g_v h$. Since g is irreducible, it follows that either $g \mid g_u$ or $g \mid h$ and either $g \mid g_v$ or $g \mid h$. Since $\deg g_u < \deg g$, we have that $g \mid g_u$ if and only if $g_u = 0$, and similarly for g_v. Now, notice that since $\deg g > 0$, at least one between g_u and g_v is not the zero polynomial and hence, $g \mid h$.

Now, let us suppose that $t > 1$ and that g divides f and all its derivatives up to the order of $t-1$. In particular, g divides all derivatives of order $t-2$ and their first partial derivatives. Thus, by the previous part of the proof, it follows that g^2 divides all partial derivatives of f of order $t-2$. By iterating this argument, we have that g^a divides all partial derivatives of f of order $t-a$ and hence, $g^t \mid f$. □

Lemma 6. *Notation as in Section 3.2. Let $t, n \in \mathbb{N}$, $t \geq 1, n \geq 0$, and let $y_{n+1}^t \mid f_{n+1}$. Then, if r is the lowest common denominator of $\sum_{i=0}^{n} c_i x^{r_i + r_{i-1} + \cdots + r_0}$, we have that $\mathcal{R} = (T^r, \sum_{i=0}^{n} c_i T^{r(r_i + r_{i-1} + \cdots + r_0)})$ is a branch of $\mathcal{C} : f = 0$ at O. Moreover, if $g(x,y) = 0$ is an equation of the parametric curve $\mathcal{R}$, then $g^t \mid f$ in $\mathbb{C}[x,y]$.*

Proof. We use the same notation introduced in the proof of Lemma 4.

By assumption $y_{n+1}^t \mid f_{n+1}$, hence, by Lemma 4, we have that there exists $h \in \mathbb{C}\{\{x\}\}[y]$ such that $f = h \cdot (y - \sum_{i=0}^n c_i x^{r_i + r_{i-1} + \cdots + r_0})^t$. In particular,

$$f(x, \sum_{i=0}^n c_i x^{r_i + r_{i-1} + \cdots + r_0}) = 0$$

and hence, $\mathcal{R}$ is a branch of $\mathcal{C}$ at O.

Since $\mathcal{R}$ has a polynomial parameterization, it is an algebraic rational curve and therefore, there exists $g \in \mathbb{C}[x,y]$, g irreducible, such that $\mathcal{R} : g = 0$. In order to prove that $g^t \mid f$ in $\mathbb{C}[x,y]$, we start proving that $g^t \mid f$ in $\mathbb{C}\{\{x\}\}[y]$. Notice that, by the proof of Lemma 4, it follows that there exists $m \in \mathbb{N}$ such that $h, (y - \sum_{i=0}^n c_i x^{r_i + r_{i-1} + \cdots + r_0}) \in \mathbb{C}[x^{\frac{1}{m}}, y]$. As a consequence, we can prove that $g^t \mid f$ in $\mathbb{C}[x^{\frac{1}{m}}, y]$. To do this, we apply Lemma 5 to $f, g \in \mathbb{C}[x^{\frac{1}{m}}, y]$. Let

$$P = (\overline{T}^r, \sum_{i=0}^n c_i \overline{T}^{r(r_i + r_{i-1} + \cdots + r_0)}) \in \mathcal{R},$$

then

$$f(P) = f(\overline{T}^r, \sum_{i=0}^n c_i \overline{T}^{r(r_i + r_{i-1} + \cdots + r_0)}) = 0,\ \forall P \in \mathcal{R} \ \Rightarrow\ P \in \mathcal{C}\ \forall P \in \mathcal{R} \ \Rightarrow\ g \mid f.$$

In order to calculate the partial derivatives of f with respect to $x^{\frac{1}{m}}$ and y, that we denote respectively by $f_{x^{\frac{1}{m}}}$ and f_y , we set $a_i := mr_i \in \mathbb{Z}$ and rewrite f as follows

$$f(x^{\frac{1}{m}}, y) = (y - \sum_{i=0}^n c_i (x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0})^t \cdot h(x^{\frac{1}{m}}, y)$$

so that the first-order partial derivates of f are

$$f_{x^{\frac{1}{m}}} = -t(\sum_{i=0}^n c_i (a_i + a_{i-1} + \cdots + a_0)(x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0 - 1})(y - \sum_{i=0}^n c_i (x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0})^{t-1} h(x^{\frac{1}{m}}, y) +$$

$$+(y - \sum_{i=0}^n c_i (x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0})^t h_{x^{\frac{1}{m}}}(x^{\frac{1}{m}}, y)$$

and

$$f_y = t(y - \sum_{i=0}^n c_i (x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0})^{t-1} h(x^{\frac{1}{m}}, y) + (y - \sum_{i=0}^n c_i (x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0})^t h_y(x^{\frac{1}{m}}, y).$$

Therefore, we have

$$f_{x^{\frac{1}{m}}}(P) = f_y(P) = 0 \ \ \forall P \in \mathcal{R} \Rightarrow g \mid f_{x^{\frac{1}{m}}} \text{ and } g \mid f_y.$$

In the same way, each partial derivative up to the order of $t-1$ is the sum of products that have at least one factor $(y - \sum_{i=0}^n c_i (x^{\frac{1}{m}})^{a_i + a_{i-1} + \cdots + a_0})$, thus, all of them vanish in each $P \in \mathcal{R}$. As a consequence, by Lemma 5, we have that $g^t \mid f$ in $\mathbb{C}[x^{\frac{1}{m}}, y]$, that is, there exists $p \in \mathbb{C}[x^{\frac{1}{m}}, y]$ such that $f = g^t p$. Since $f, g \in \mathbb{C}[x,y] \subseteq \mathbb{C}[x^{\frac{1}{m}}, y]$, in order to show that $g^t \mid f$ in $\mathbb{C}[x,y]$ it is enough to show that $p \in \mathbb{C}[x,y]$. Let us write p as $p = p_1 + p_2$, where each monomial of p_1 is in $\mathbb{C}[x,y]$ and each monomial of p_2 is in $\mathbb{C}[x^{\frac{1}{m}}, y] \setminus \mathbb{C}[x,y]$. We have that

$$g^t p_2 = f - g^t p_1$$

so that, since $f, g, p_1 \in \mathbb{C}[x, y]$, also $g^t p_2 \in \mathbb{C}[x, y]$. Now, we observe that if $x^a y^b$ is a monomial of $g^t p_2$ then a is sum of an integer, coming from a monomial of g^t, and of a rational that is not an integer, coming from a monomial of p_2, and thus a is a rational but not an integer. As a consequence, we have that $g^t p_2 = 0$. Hence, since $\mathbb{C}[x^{\frac{1}{m}}, y]$ is a domain and $g \neq 0$, we have that $p_2 = 0$ and $p = p_1 \in \mathbb{C}[x, y]$ and this ends the proof. □

3.4. The Algorithm

The Newton–Puiseux Algorithm

Now, we present the Newton–Puiseux Algorithm based on the procedures described and justified in the above sections. Notice that we suppose $x \nmid f$, since we used this hypothesis in the $*$-procedure; of course, if one wants to study a curve that has the y-axis as a component, it is sufficient to remove the factor x from f and then to add the vertical branch $\{x = 0\}$ (parameterized by $(0, T)$) to the branches at O obtained by the Algorithm.

We gave Algorithm 1 for reduced curves; notice that when f is not irreducible, it can be much easier, computationally, to work separately on its irreducible components (if they are known, to find them is not computationally easy, e.g., see [18], where an algorithm is given) and then consider the union of the branches of any such component.

Algorithm 1 Study of the branches at $(0, 0)$ of a reduced plane curve $\mathcal{C}$ of equation $f = \sum_{i+j=1}^{d} a_{ij} x^i y^j$, $a_{ij} \in \mathbb{C}$, such that $x \nmid f$

Input: $\mathcal{C}$: $f = \sum_{i+j=1}^{d} a_{ij} x^i y^j$, $O = (0, 0)$.

Output: Integers $r_1, \ldots, r_s$ and $p_1 \in \mathbb{C}[x^{\frac{1}{r_1}}], \ldots, p_s \in \mathbb{C}[x^{\frac{1}{r_s}}]$ such that $\mathcal{C}$ possesses s branches at O, and each branch is approximated by $(T^{r_i}, p_i(T^{r_i}))$.

```
1: Apply the $*$-procedure to $f$ and go on with the $f^{(i_1,j_1)\cdots(i_n,j_n)}$'s thus obtained. Form
   the graph $\mathbf{G}_f$, applying the Stop criterion.
2: For any descending path $\gamma$ in $\mathbf{G}_f$, work as in 3.2, until you obtain a $p_\gamma(x) \in \mathbb{C}[x^{\frac{1}{r_\gamma}}]$,
   which yields an approximation of a branch of $\mathcal{C}$ at $O$.
3: Consider all $p_\gamma$'s obtained at the previous step and find the ones that give equivalent
   parameterizations. If $p_\gamma$ and $p_{\gamma'}$ are the truncations of Puiseux series $\tilde{p}_\gamma$ and $\tilde{p}_{\gamma'}$, they
   give equivalent parameterizations if and only if they have the same lowest common
   denominator $r$ and it exists $\omega_r \in \mathbb{C}$ with $\omega_r^r = 1$, such that $\tilde{p}_\gamma(x^{\frac{1}{r}}) = \tilde{p}_{\gamma'}(\omega_r x^{\frac{1}{r}})$ up to
   $\overline{n}$, where $\overline{n} = \min\{n \in \mathbb{N} \mid f_n(x, y_n)$ and $f'_n(x, y_n)$ satisfy Theorem 1$\}$.
   Keep only one of the $p_\gamma$'s for each class of equivalence.
4: The number $s$ of equivalence classes found at step 3 is the number of branches of
   $\mathcal{C}$ at $O$. The data $r_1, \ldots, r_s$ and $p_1 \in \mathbb{C}[x^{\frac{1}{r_1}}], , \ldots, p_s \in \mathbb{C}[x^{\frac{1}{r_s}}]$ obtained before are the
   required output, and $(T^{r_i}, p_i(T^{r_i}))$, $i = 1, \ldots, s$ give approximations of the $s$ branches
   of our curve. END
```

We summarize this (quite obvious) procedure in the next algorithm, where we also drop the "f reduced" hypothesis, by adding that any branch must be counted as "n-ple" if obtained from a factor f_i^n.

Notice that also in Algorithm 2, we suppose that $x = 0$ is not a component of $\mathcal{C}$; obviously, if there is a factor of f which is x^n, we just remove it before applying the Algorithm and then we add the branch $\{x = 0\}$ counted n times to the output.

Algorithm 2 Study of the branches at $(0,0)$ of a plane curve $\mathcal{C}$ of equation $f = \sum_{i+j=1}^{d} a_{ij}x^iy^j$, $a_{ij} \in \mathbb{C}$, $f = f_1^{n_1} \dots f_m^{n_m}$ (the irreducible decomposition of f), with $f_\ell \neq x$, $\ell = 1,\dots,m$

Input: $\mathcal{C}$: $f = \sum_{i+j=1}^{d} a_{ij}x^iy^j = f_1^{n_1} \dots f_m^{n_m}$, $O = (0,0)$.

Output: Integers $r_{\ell,1},\dots,r_{\ell,s_\ell}$ and $p_{\ell,1} \in \mathbb{C}[x^{\frac{1}{r_{\ell,1}}}],\dots,p_{\ell,s_\ell} \in \mathbb{C}[x^{\frac{1}{r_{\ell,s_\ell}}}]$, $\ell = 1,\dots,m' \leq m$, such that $\mathcal{C}$ possesses $s = s_1 + \dots + s_{m'}$ branches at O, and each branch is approximated by $(T^{r_{\ell,i}}, p_{\ell,i}(T^{r_{\ell,i}}))$, $i \in \{1,\dots,s_\ell\}$.

```
For each $f_\ell$, $\ell = 1,\dots,m$, consider $f_\ell(O)$; let (up to reordering the $f_\ell$'s) $f_1(O) = \dots = f_{m'}(O) = 0$, while $f_\ell(O) \neq 0$ for $\ell > m'$.
Apply Algorithm 1 to any $f_\ell$, $1 \leq \ell \leq m'$, so to get $p_{\ell,1} \in \mathbb{C}[x^{\frac{1}{r_{\ell,1}}}],\dots,p_{\ell,s_\ell} \in \mathbb{C}[x^{\frac{1}{r_{\ell,s_\ell}}}]$ such that $(T^{r_{\ell,i}}, p_{\ell,i}(T^{r_{\ell,i}}))$ are the (non-equivalent) approximations of the $s_\ell$ branches of $\mathcal{C}_\ell : \{f_\ell = 0\}$ at $O$.
Consider all $(T^{r_{\ell,i}}, p_{\ell,i}(T^{r_{\ell,i}}))$, $i \in \{1,\dots,s_\ell\}$, $\ell \in \{1,\dots,m'\}$ obtained in the previous step. They give the approximations of all the branches of $\mathcal{C}$ at $O$; each of them must be considered $n_\ell$ times.
END
```

3.5. An Example

The following example illustrates how the Algorithm works.

Example 3. Consider the integral curve $\mathcal{C} : f = 0$, where

$$f = 2y^6 + 6xy^5 - 8x^3y^3 + 2x^3y^4 + (2\sqrt{3}+2)x^4y^3 + (4\sqrt{3}-4)x^5y^2 + (\sqrt{3}-2)x^7y + \frac{\sqrt{3}-2}{8}x^{10} + 2x^{11}.$$

We have $m_O(\mathcal{C}) = 6$. The Newton polygon $\Gamma(f_0) = \Gamma(f)$ is given in Figure 3:

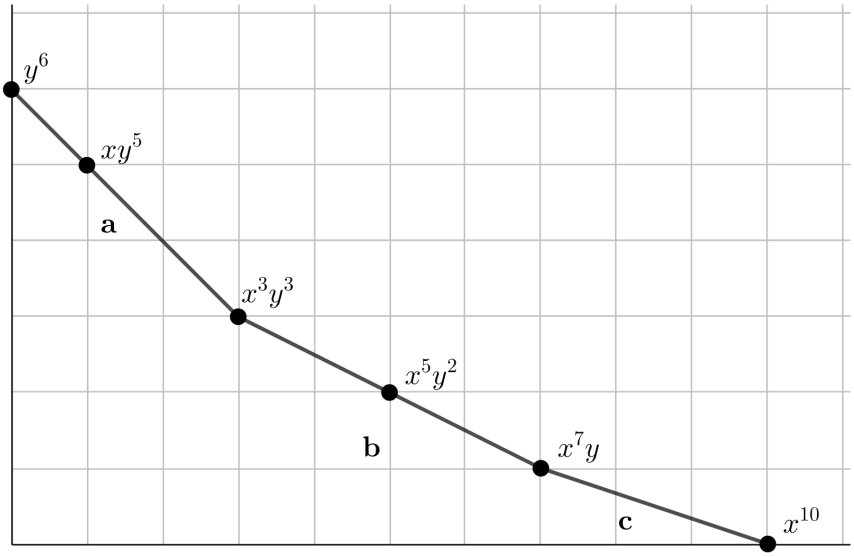

Figure 3. The Newton Polygon of f_0.

We start setting $f_0 = f$. Let us analyze each possible choice individually:

- $\mathbf{a}_0 = \mathbf{a}$
  We have

$$f_0^{\mathbf{a}_0} = f^{\mathbf{a}_0} = 2y^6 + 6xy^5 - 8x^3y^3$$

$$g_0 = \frac{f_0^{\mathbf{a}_0}}{y^3} = 2(y^3 + 3xy^2 - 4x^3) = 2(y+2x)^2(y-x).$$

Thus, we have two distinct roots of g_0, namely, $q_{01} = -2x$ double and $q_{02} = x$ simple, and, as a consequence, two possible choice for q_0:

- $q_0 = q_{01}$

  We have

$$f_0(x, x(-2+y_1)) = x^6(48y_1^2 - 88y_1^3 + 60y_1^4 - 18y_1^5 + 2y_1^6 + (8\sqrt{3} - 24)xy_1 + (-8\sqrt{3} + 32)xy_1^2 + \\ +(2\sqrt{3} - 14)xy_1^3 + 2xy_1^4 + (-2\sqrt{3} + 4)x^2 + (\sqrt{3} - 2)x^2y_1 + (\frac{\sqrt{3}}{8} - \frac{1}{4})x^4 + 2x^5)$$

hence,

$$f_1(x, y_1) = \frac{f_0(x, x(-2+y_1))}{x^6} = 48y_1^2 - 88y_1^3 + 60y_1^4 - 18y_1^5 + 2y_1^6 + (8\sqrt{3} - 24)xy_1 + (-8\sqrt{3} + 32)xy_1^2 + \\ +(2\sqrt{3} - 14)xy_1^3 + 2xy_1^4 + (-2\sqrt{3} + 4)x^2 + (\sqrt{3} - 2)x^2y_1 + (\frac{\sqrt{3}}{8} - \frac{1}{4})x^4 + 2x^5.$$

The Newton polygon $\Gamma(f_1)$ is given in Figure 4

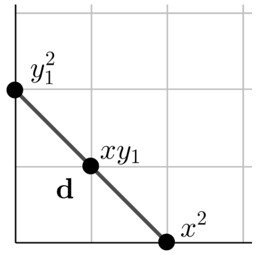

Figure 4. Newton Polygon of f_1.

We have an unique edge $\mathbf{a}_1 = \mathbf{d}$, therefore, we obtain

$$g_1 = f_1^{\mathbf{a}_1} = 48y_1^2 + (8\sqrt{3} - 24)xy_1 + (-2\sqrt{3} + 4)x^2 = (4\sqrt{3}y + (-\sqrt{3} + 1)x)^2.$$

There is an unique root of g_1, which is

$$q_1 = -\frac{-\sqrt{3} + 1}{4\sqrt{3}}x = \frac{3 - \sqrt{3}}{12}x.$$

Thus, we have

$$f_1(x, x(\frac{3-\sqrt{3}}{12} + y_2)) = x^2(48y_2^2 + (\frac{-\sqrt{3}}{18} + \frac{1}{6})x + (\frac{8\sqrt{3}}{3} - 4)xy_2 + (-\frac{\sqrt{3}}{72} - \frac{1}{16})x^2 + (14\sqrt{3} - 34)xy_2^2 + \\ +(-\frac{23\sqrt{3}}{12} + \frac{13}{4})x^2y_2 + (\frac{41\sqrt{3}}{1152} + \frac{2233}{1152})x^3 - 88xy_2^3 + (-10\sqrt{3} + 18)x^2y_2^2 + (\frac{35\sqrt{3}}{72} - \frac{27}{32})x^3y_2 + \\ +(-\frac{5\sqrt{3}}{2304} + \frac{13}{3456})x^4 + (-18\sqrt{3} + 46)x^2y_2^3 + (\frac{21\sqrt{3}}{8} - \frac{37}{8})x^3y_2^2 + (-\frac{19\sqrt{3}}{576} + \frac{11}{192})x^4y_2 + 60x^2y_2^4 + \\ +(\frac{41\sqrt{3}}{6} - 13)x^3y_2^3 + (-\frac{5\sqrt{3}}{24} + \frac{35}{96})x^4y_2^2 + (\frac{15\sqrt{3}}{2} - \frac{41}{2})x^3y_2^4 + (-\frac{25\sqrt{3}}{36} + \frac{5}{4})x^4y_2^3 - 18x^3y_2^5 + \\ +(-\frac{5\sqrt{3}}{4} + \frac{5}{2})x^4y_2^4 + (-\sqrt{3} + 3)x^4y_2^5 + 2x^4y_2^6)$$

$$f_2(x, y_2) = \frac{f_1(x, x(\frac{3-\sqrt{3}}{12} + y_2))}{x^2}.$$

The Newton polygon $\Gamma(f_2)$ is given in Figure 5.

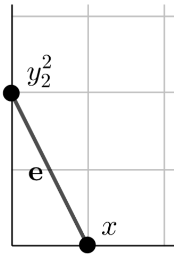

Figure 5. Newton Polygon of f_2.

We have an unique edge $\mathbf{a}_2 = \mathbf{e}$ and we obtain:

$$g_2 = f_2^{\mathbf{a}_2} = 48y_2^2 + (\frac{-\sqrt{3}}{18} + \frac{1}{6}x)$$

whose roots are

$$q_{21} = i\sqrt{\frac{3-\sqrt{3}}{864}}x^{\frac{1}{2}} \qquad q_{22} = -i\sqrt{\frac{3-\sqrt{3}}{864}}x^{\frac{1}{2}}.$$

By Lemma 3, choosing either $q_2 = q_{21}$ or $q_2 = q_{22}$, the polynomial f_3 satisfies the hypothesis of Theorem 1; hence, by 3.1, we can stop. Since

$$r_0 = 1, \quad r_1 = 1, \quad r_2 = \frac{1}{2},$$

we obtain

$$p_1 = -2x + \frac{3-\sqrt{3}}{12}x^2 + i\sqrt{\frac{3-\sqrt{3}}{864}}x^{\frac{5}{2}} + \ldots$$

$$p_2 = -2x + \frac{3-\sqrt{3}}{12}x^2 - i\sqrt{\frac{3-\sqrt{3}}{864}}x^{\frac{5}{2}} + \ldots.$$

Therefore, we have the following parameterizations

$$R_1 = (T^2, -2T^2 + \frac{3-\sqrt{3}}{12}T^4 + i\sqrt{\frac{3-\sqrt{3}}{864}}T^5 + \ldots)$$

$$R_2 = (T^2, -2T^2 + \frac{3-\sqrt{3}}{12}T^4 - i\sqrt{\frac{3-\sqrt{3}}{864}}T^5 + \ldots).$$

We have that $p_1(x) = p_2(-x)$, at O, so by Step 3 of the Algorithm with $\omega_2 = -1$, we conclude that R_1 and R_2 are equivalent parameterizations of the same 2-branch of $\mathcal{C}$.

- $q_0 = q_{02}$

We have

$$f_0(x, x(1+y_1)) = x^6(18y_1 + 6\sqrt{3}x + 66y_1^2 + (14\sqrt{3}+6)xy_1 + (\sqrt{3}-2)x^2 + 92y_1^3 + (10\sqrt{3}+14)xy_1^2 +$$

$$+(\sqrt{3}-2)x^2y_1 + 60y_1^4 + (2\sqrt{3}+10)xy_1^3 + (\frac{\sqrt{3}}{8} - \frac{1}{4})x^4 + 18y_1^5 + 2xy_1^4 + 2x^5 + 2y_1^6)$$

and hence,

$$f_1(x, y_1) = \frac{f_0(x, x(1+y_1))}{x^6} = 18y_1 + 6\sqrt{3}x + 66y_1^2 + (14\sqrt{3}+6)xy_1 + (\sqrt{3}-2)x^2 + 92y_1^3 +$$

$$+(10\sqrt{3}+14)xy_1^2 + (\sqrt{3}-2)x^2y_1 + 60y_1^4 + (2\sqrt{3}+10)xy_1^3 + (\frac{\sqrt{3}}{8} - \frac{1}{4})x^4 + 18y_1^5 + 2xy_1^4 + 2x^5 + 2y_1^6).$$

The Newton polygon $\Gamma(f_1)$ is given in Figure 6.

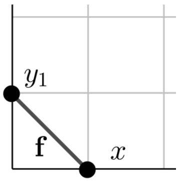

Figure 6. Newton Polygon of f_1 (Case $q_0 = q_{02}$).

We can just choose $\mathbf{a}_1 = \mathbf{f}$ obtaining

$$g_1 = f_1^{\mathbf{f}} = 18y_1 + 6\sqrt{3}x$$

whose unique root is

$$q_1 = -\frac{\sqrt{3}}{3}x.$$

Moreover, f_1 satisfies the hypothesis of Theorem 1, therefore, by Remark 3.1, we can stop by obtaining the following root of f

$$p_3(x) = x - \frac{\sqrt{3}}{3}x^2 + \dots.$$

Hence, we have the parameterization

$$R_3 = (T, T - \frac{\sqrt{3}}{3}T^2 + \dots)$$

which is a 1-branch of $\mathcal{C}$ at O.

- $\mathbf{a}_0 = \mathbf{b}$
  We have

$$f_0^{\mathbf{a}_0} = f^{\mathbf{a}_0} = -8x^3y^3 + (4\sqrt{3}-4)x^5y^2 + (\sqrt{3}-2)x^7y$$

$$g_0 = \frac{f_0^{\mathbf{a}_0}}{x^3y} = -8y^2 + (4\sqrt{3}-4)x^2y^2 + (\sqrt{3}-2)x^4 = -(2\sqrt{2}y + \frac{\sqrt{2}-\sqrt{6}}{2}x^2)^2.$$

  Hence, we have only one choice for q_0, that is,

$$q_0 = -\frac{\sqrt{2}-\sqrt{6}}{4\sqrt{2}}x^2 = \frac{\sqrt{3}-1}{4}x^2.$$

  Proceeding, we obtain

$$f_0(x, x^2(\frac{\sqrt{3}-1}{4} + y_1)) = x^9((-2\sqrt{3}+2)y_1^2 + (\frac{3\sqrt{3}}{4} - \frac{3}{4})xy_1 + (\frac{17\sqrt{3}}{128} + \frac{227}{128})x^2 - 8y_1^3 + 3xy_1^2 +$$

$$+(-\frac{9\sqrt{3}}{8} + \frac{65}{32})x^2y_1 + (\frac{-15\sqrt{3}}{256} + \frac{13}{128})x^3 + (2\sqrt{3}+2)xy_1^3 + (\frac{33\sqrt{3}}{8} - \frac{51}{8})x^2y_1^2 + (\frac{33\sqrt{3}}{64} - \frac{57}{64})x^3y_1 +$$

$$+(-\frac{11\sqrt{3}}{2} + 13)x^2y_1^3 + (-\frac{15\sqrt{3}}{8} + \frac{105}{32})x^3y_1^2 + (\frac{15\sqrt{3}}{2} - \frac{11}{2})x^2y_1^4 + (\frac{15\sqrt{3}}{4} - \frac{25}{4})x^3y_1^3 + 6x^2y_1^5 +$$

$$+(-\frac{15\sqrt{3}}{4}+\frac{15}{2})x^3y_1^4+(3\sqrt{3}-3)x^3y_1^5+2x^3y_1^6)$$

$$f_1(x,y_1)=\frac{f_0(x,x^2(\frac{\sqrt{3}-1}{4}+y_1))}{x^9}.$$

The Newton Polygon $\Gamma(f_1)$ is given in Figure 7:

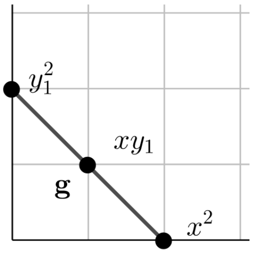

Figure 7. The Newton Polynomial of $\frac{f_0(x,x^2(\frac{\sqrt{3}-1}{4}+y_1))}{x^9}$.

We can only choose $\mathbf{a}_1=\mathbf{g}$, obtaining

$$g_1=f_1^{\mathbf{g}}=(-2\sqrt{3}+2)y_1^2+(\frac{3\sqrt{3}}{4}-\frac{3}{4})xy_1+(\frac{17\sqrt{3}}{128}+\frac{227}{128})x^2$$

whose roots are

$$q_{11}=\frac{3-3\sqrt{3}+2\sqrt{35+48\sqrt{3}}}{16(-\sqrt{3}+1)}x$$

$$q_{12}=\frac{3-3\sqrt{3}-2\sqrt{35+48\sqrt{3}}}{16(-\sqrt{3}+1)}x.$$

By Lemma 3, choosing either $q_1=q_{11}$ or $q_1=q_{12}$, the polynomial f_2 satisfies the hypothesis of Theorem 1. Hence, by Remark 3.1, we can stop obtaining the following roots of f

$$p_4=\frac{\sqrt{3}-1}{4}x^2+\frac{3-3\sqrt{3}+2\sqrt{35+48\sqrt{3}}}{16(-\sqrt{3}+1)}x^3+\dots$$

$$p_5=\frac{\sqrt{3}-1}{4}x^2+\frac{3-3\sqrt{3}-2\sqrt{35+48\sqrt{3}}}{16(-\sqrt{3}+1)}x^3+\dots.$$

Hence we have the parameterizations

$$R_4=(T,\frac{\sqrt{3}-1}{4}T^2+\frac{3-3\sqrt{3}+2\sqrt{35+48\sqrt{3}}}{16(-\sqrt{3}+1)}T^3+\dots)$$

$$R_5=(T,\frac{\sqrt{3}-1}{4}T^2+\frac{3-3\sqrt{3}-2\sqrt{35+48\sqrt{3}}}{16(-\sqrt{3}+1)}T^3+\dots)$$

that give two distinct 1-branches of $\mathcal{C}$ at O.

- $\mathbf{a}_0=\mathbf{c}$
  We have

$$f_0^{\mathbf{c}}=f^{\mathbf{c}}=(\sqrt{3}-2)x^7y+\frac{\sqrt{3}-2}{8}x^{10}$$

$$g_0=\frac{f_0^{\mathbf{c}}}{x^7}=(\sqrt{3}-2)y+\frac{\sqrt{3}-2}{8}x^3.$$

We have just a choice for q_0, that is,

$$q_0=-\frac{1}{8}x^3.$$

Proceeding, we obtain

$$f_0(x, x^3(-\frac{1}{8}+y_1)) = x^{10}((\sqrt{3}-2)y_1 + (\frac{\sqrt{3}}{16}+\frac{31}{16}))x + (-\sqrt{3}+1)xy_1 + \frac{1}{64}x^2 + (4\sqrt{3}-4)xy_1^2 - \frac{3}{8}x^2y_1 +$$
$$+(-\frac{\sqrt{3}}{256}-\frac{1}{256})x^3 + 3x^2y_1^2 + (\frac{3\sqrt{3}}{32}+\frac{3}{32})x^3y_1 - 8x^2y_1^3 + (-\frac{3\sqrt{3}}{4}-\frac{3}{4})x^3y_1^2 + \frac{1}{2048}x^5 + (2\sqrt{3}+2)x^3y_1^3 +$$
$$-\frac{1}{64}x^5y_1 - \frac{3}{16384}x^6 + \frac{3}{16}x^5y_1^2 + \frac{15}{2048}x^6y_1 - x^5y_1^3 - \frac{15}{128}x^6y_1^2 + \frac{1}{131072}x^8 + 2x^5y_1^4 + \frac{15}{16}x^6y_1^3 - \frac{3}{8192}x^8y_1 +$$
$$-\frac{15}{4}x^6y_1^4 + \frac{15}{2048}x^8y_1^2 + 6x^6y_1^5 - \frac{5}{64}x^8y_1^3 + \frac{15}{32}x^8y_1^4 - \frac{3}{2}x^8y_1^5 + 2x^8y_1^6$$

$$f_1(x, y_1) = \frac{f_0(x, x^3(-\frac{1}{8}+y_1))}{x^{10}}.$$

The Newton polygon $\Gamma(f_1)$ is given in Figure 8.
We can only choose $\mathbf{a}_1 = \mathbf{h}$, obtaining

$$g_1 = f_1^{\mathbf{h}} = (\sqrt{3}-2)y_1 + (\frac{\sqrt{3}}{16}+\frac{31}{16})x$$

whose unique root is

$$q_1 = -\frac{33\sqrt{3}+65}{16}x.$$

Moreover, f_1 satisfies the hypothesis of Theorem 1, therefore, by Remark 3.1, we can stop obtaining the following root of f

$$p_6 = -\frac{1}{8}x^3 - \frac{33\sqrt{3}+65}{16}x^4 + \ldots$$

which gives the parameterization

$$R_6(T, -\frac{1}{8}T^3 - \frac{33\sqrt{3}+65}{16}T^4 + \ldots)$$

which is a 1-branch of $\mathcal{C}$ at O.

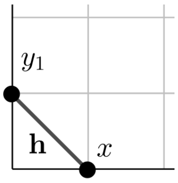

Figure 8. The Newton polygon of $\frac{f_0(x,x^3(-\frac{1}{8}+y_1))}{x^{10}}$.

We have studied all possible choices so we can stop the analysis.

Since we saw that R_1 and R_2 are equivalent parameterizations of the same 2-branch of $\mathcal{C}$, we have found five branches of $\mathcal{C}$, see Figure 9.

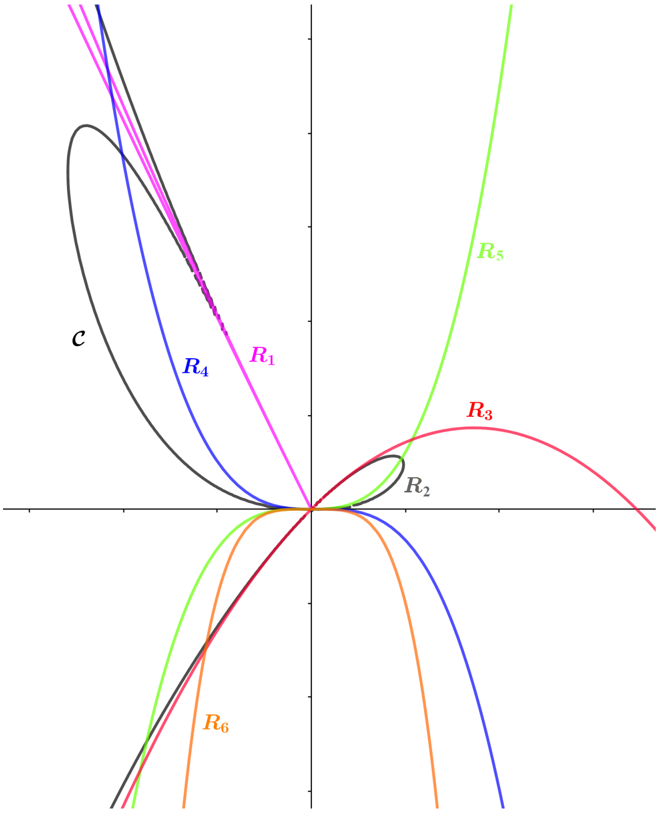

Figure 9. The six branches of the curve at O.

4. Triple Points

4.1. Output of the Newton–Puiseux Algorithm for Triple Points

In this section, we use the Newton–Puiseux algorithm to study triple points for a reduced algebraic plane curve. Let $\mathcal{C}$: $f = 0$ be a plane curve of degree d and let P be a triple point for $\mathcal{C}$ (i.e., $m_P(\mathcal{C}) = 3$).

We recall that, by Remark 3, the curve $\mathcal{C}$ can have three 1-branches, or one 2-branch and one 1-branch, or a 3-branch at P. Anyway, if the tangent cone is made up of three distinct tangent lines, P is an ordinary singularity, and it is analytically equivalent to $y(y^2 - x^2)$, while if there are two distinct tangents, the singularity is analytically equivalent to $y(y^2 - x^n)$, $n \geq 3$, see [12].

Hence, we focus on the most interesting case, i.e., when $\mathcal{C}$ has a unique triple tangent r at P (for an analysis modulo analytic equivalence of the branches at a triple point, see [HH]). It is not restrictive to assume that $P = O$ and that r is given by $y = 0$, hence,

$$f = y^3 + \sum_{i+j=4}^{d} a_{ij} x^i y^j.$$

The cases where f is divisible by y are trivial, so we can assume that $y \nmid f$, that is, f is Newton-convenient. Notice that if $\mathcal{C}$ has one 3-branch at O, then its parameterization will be of type $(T^3, \sum_{n\geq 4} a_n T^n)$ (this follows by Definition 3, since the unique tangent to the branch is $y = 0$).

We use the notation of Section 3.2, with the unique exception that the g_n associated to $f_n^{\mathbf{a}_n}$ will be denoted by $g_n^{\mathbf{a}_n}$, in order to specify which edge we refer to. Notice that if $\mathbf{a}_n$ is an edge of the Newton polygon, the height of $\mathbf{a}_n$ is denoted by $h(\mathbf{a}_n)$, while the height of the Newton polygon, denoted by $h(\Gamma(f_n))$, is defined to be the sum of the heights of its edges.

We indicate with ${}_n a_{ij}$ the $x^i y_n^j$ coefficient in f_n. We also denote by A_{c_n,r_n} the subalgorithm of Algorithm 1 starting inside step n with the choice of the root $q_n = c_n x^{r_n}$ for g_n.

In order to describe the output of the algorithm, we have to answer the following questions:

1. Which are the possible choices of c_n and r_n in each case?
2. When, for each choice of (c_n, r_n), does the algorithm A_{c_n,r_n} stop?

The Stop Criterion 3.1 is an answer to question 2, but we can improve on it:

Remark 5. By Lemma 3, if c_n has multicipliy 1 as a root of g_n, then f_{n+1} satisfies the hypothesis of Theorem 1, because in f_{n+1}, there is the monomial $c_n y_{n+1}$. Hence, if c_n has multiplicity 1 as a root of g_n, A_{c_n,r_n} stops at the step $n-1$, i.e., without calculating f_{n+1}. From now on, we use this new Stop criterion because it is more efficient.

To answer question 1, we observe that the following statements hold:

(i) Since the height of the Newton Polygon decreases at each step, i.e., $h(\Gamma(f_{n+1})) \leq h(\Gamma(f_n))$, and $h(\Gamma(f)) = 3$, we have to consider Newton Polygons of height ≤ 3 only;
(ii) In general, $f_n \in \mathbb{C}\{\{x\}\}[y]$. However, since $f \in \mathbb{C}[x,y]$, there exists $\overline{n} \in \mathbb{N}$ such that $f_n \in \mathbb{C}[x,y] \ \ \forall \ \ n \leq \overline{n}$. Hence, for $n \leq \bar{n}$, the vertices of $\Gamma(f_n)$ are in $\mathbb{N}^2$.

We start answering question 1 just for $n \leq \bar{n}$; we shall see later (see Remark 7) that this is enough.

Remark 6. The possible shapes of the Newton Polygons arising when applying our algorithm in the case of a triple point with a triple tangent and for $n \leq \overline{n}$ are described in the Figure 10, and each of the corresponding cases is discussed below. Namely, in each case, we write explicitly the g_n associated with each edge and its roots $c_n x^{r_n}$ with their relative multiplicities, so that we have a complete answer to question 1 first; after that, we examine the algorithms A_{c_n,r_n} for each (c_n, r_n).

Case 1: $\Gamma(f_n) = \{\mathbf{a}\}$, $h(\mathbf{a}) = 1$, and we have

$$g_n^{\mathbf{a}} = {}_na_{01} y_n + {}_na_{i_1 0} x^{i_1}$$

$g_n^{\mathbf{a}}$ has a unique root of multiplicity 1, $c_1 x^{i_1}$ with $c_1 = -\frac{{}_na_{i_1 0}}{{}_na_{01}}$, which by Remark 5, gives a 1-branch of $\mathcal{C}$, and the algorithm A_{c_1,i_1} stops here.

Case 2.1: $\Gamma(f_n) = \{\mathbf{a}\}$, $h(\mathbf{a}) = 2$, and $i_1 \not\equiv 0 \pmod 2$, so we have

$$g_n^{\mathbf{a}} = {}_na_{02} y_n^2 + {}_na_{i_1 0} x^{i_1};$$

$g_n^{\mathbf{a}}$ has two distinct roots of multiplicity 1 each:

$$c_1 x^{\frac{i_1}{2}}, \quad c_2 x^{\frac{i_1}{2}} \qquad \text{where } c_1^2 = -\frac{{}_na_{i_1 0}}{{}_na_{02}}, \ \ c_2 = -c_1.$$

By Remark 5, the algorithms $A_{c_1,\frac{i_1}{2}}$ and $A_{-c_1,\frac{i_1}{2}}$ stop here; moreover, they give equivalent parameterizations of the same 2-branch of $\mathcal{C}$, since, if that were not true, the sum of the multiplicities of the branches at O would be ≥ 4. The parameterization $(T^2, \dots + c_1 T^{i_1} + \dots)$, composed with the biolomorphic function $T \mapsto -T$, is equal to $(T^2, \dots + c_2 T^{i_1} + \dots)$; in fact, since they are equivalent, by Remark 1, there is an ω, $\omega^2 = 1$, such that R_2 is obtained from R_1 by substituting $T \to \omega T$; since $c_1(-T)^{i_1} = c_2 T^{i_1}$, we obtain $\omega = -1$.

Case 2.2: $\Gamma(f_n) = \{\mathbf{a}\}$, $h(\mathbf{a}) = 2$, and $i_1 \equiv 0 \pmod 2$, so we have

$$g_n^{\mathbf{a}} = {}_na_{02} y_n^2 + {}_na_{\frac{i_1}{2}1} x^{\frac{i_1}{2}} y_n + {}_na_{i_1 0} x^{i_1}.$$

Let $c_1 x^{\frac{i_1}{2}}, c_2 x^{\frac{i_1}{2}}$ denote the roots of $g_n^{\mathbf{a}}$; we distinguish two cases:

Case 2.2.1: If $c_1 \neq c_2$, i.e., if $g_n^{\mathbf{a}}$ has two roots of multiplicity 1, they give two distinct 1-branches of $\mathcal{C}$ and the algorithms $A_{c_1,\frac{i_1}{2}}$ and $A_{c_2,\frac{i_1}{2}}$ stop here.

Notice that if ${}_n a_{\frac{i_1}{2}1} = 0$, we apparently are in the same situation as Case 2.1, but here $k := \frac{i_1}{2}$ is a positive integer, so the two simple roots are

$$c_1 x^k, \quad -c_1 x^k$$

and the biolomorphic function $T \mapsto -T$ used in 2.1 does not work here: it does not give the equivalence of $(T, \cdots + c_1 T^k + \dots)$ and $(T, \cdots - c_1 T^k + \dots)$. In fact, the two simple roots correspond to two non-equivalent parameterizations, each one for a different 1-branch.

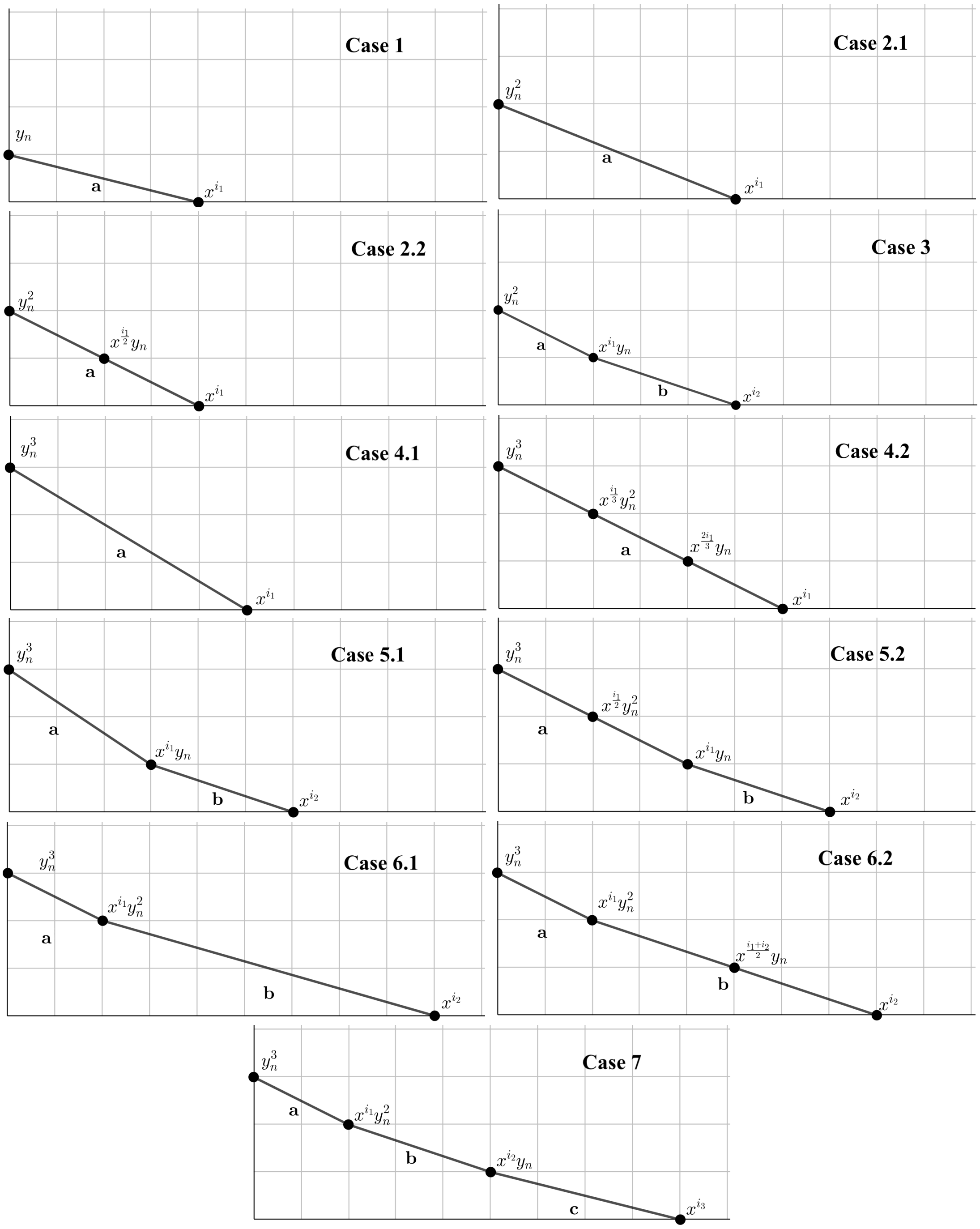

Figure 10. Newton Polygons for a triple point with a triple tangent.

Case 2.2.2: If $c_1 = c_2$, i.e., if $g_n^{\mathbf{a}}$ has one double root, the algorithm $A_{c_1,\frac{i_1}{2}}$ goes on: it can yield either two 1-branches or one 2-branch of $\mathcal{C}$, hence, globally, we may find either three 1-branches or one 2-branch and one 1-branch. Notice that f_{n+1} is still in $\mathbb{C}[x,y]$, since $i_1 \equiv 0 \pmod 2$.

Case 3: $\Gamma(f_n) = \{\mathbf{a}, \mathbf{b}\}, h(\mathbf{a}) = 1, h(\mathbf{b}) = 1$; we have

$$g_n^{\mathbf{a}} = {}_na_{02}y_n + {}_na_{i_11}x^{i_1}, \qquad g_n^{\mathbf{b}} = {}_na_{i_11}y_n + {}_na_{i_20}x^{i_2-i_1}.$$

Both $g_n^{\mathbf{a}}$ and $g_n^{\mathbf{b}}$ have a unique root of multiplicity 1, $-\frac{{}_na_{i_11}}{{}_na_{02}}x^{i_1}$ and $-\frac{{}_na_{i_20}}{{}_na_{i_11}}x^{i_2-i_1}$; each of them gives a different 1-branch of $\mathcal{C}$, and the algorithms stops here.

Case 4.1: $\Gamma(f_n) = \{\mathbf{a}\}, h(\mathbf{a}) = 3$, and $i_1 \not\equiv 0 \pmod 3$, so we have

$$g_n^{\mathbf{a}} = {}_na_{03}y_n^3 + {}_na_{i_10}x^{i_1}$$

$g_n^{\mathbf{a}}$ has three distinct roots of multiplicity 1, namely:

$$c_1x^{\frac{i_1}{3}}, \quad c_2x^{\frac{i_1}{3}}, \quad c_3x^{\frac{i_1}{3}}, \qquad \text{where } c_1^3 = -\frac{{}_na_{i_10}}{{}_na_{03}}, \;\; c_2 = e^{i\frac{2\pi}{3}}c_1, \;\; c_3 = e^{i\frac{4\pi}{3}}c_1$$

and, as in Case 2.1, all of them give equivalent parameterizations of the same 3-branch of $\mathcal{C}$, since otherwise, the sum of the multiplicities of the branches at O would be ≥ 6; for example the parameterization $(T^3, \cdots + c_1T^{i_1} + \dots)$ composed with the biolomorphic function $T \mapsto e^{i\frac{2\pi}{3}}T$ is equal to $(T^3, \cdots + c_2T^{i_1} + \dots)$. Since the three roots have multiplicity 1, the algorithms stop.

Case 4.2: $\Gamma(f_n) = \{\mathbf{a}\}, h(\mathbf{a}) = 3$, and $i_1 \equiv 0 \pmod 3$, so we have

$$g_n^{\mathbf{a}} = {}_na_{03}y_n^3 + {}_na_{\frac{i_1}{3}2}x^{\frac{i_1}{3}}y_n^2 + {}_na_{\frac{2i_1}{3}1}x^{2\frac{i_1}{3}}y_n + {}_na_{i_10}x^{i_1}.$$

Let $c_1x^{\frac{i_1}{3}}, \quad c_2x^{\frac{i_1}{3}}, \quad c_3x^{\frac{i_1}{3}}$ denote the roots of $g_n^{\mathbf{a}}$; we distinguish three cases:

Case 4.2.1: If $c_1 \neq c_2 \neq c_3 \neq c_1$, i.e., if $g_n^{\mathbf{a}}$ has three roots of multiplicity 1, they give three distinct 1-branches of $\mathcal{C}$ so the algorithms stop here.

Notice that if ${}_na_{\frac{i_1}{3}2} = {}_na_{\frac{2i_1}{3}1} = 0$, we apparently are in the same situation as Case 4.1, but here $k := \frac{i_1}{3}$ is a positive integer, the three simple roots are

$$c_1x^k, \quad e^{i\frac{2\pi}{3}}c_1x^k, \quad e^{i\frac{2\pi}{3}}c_1x^k$$

and the biolomorphic function $T \mapsto e^{i\frac{2\pi}{3}}T$ used in 4.1 does not work here: it does not give the equivalence of $(T, \cdots + c_1T^k + \dots)$ and $(T, \cdots + c_2T^k + \dots)$. In fact, the three simple roots correspond to three non-equivalent parameterizations (each of a different 1-branch).

Case 4.2.2: If $c_1 \neq c_2 = c_3$, , i.e., if $g_n^{\mathbf{a}}$ has two roots, one double and one simple; the algorithm $A_{c_1,\frac{i_1}{3}}$ stops and gives a 1-branch of $\mathcal{C}$. On the other hand, $A_{c_2,\frac{i_1}{3}}$ does not stop because the relative root has multiplicity 2; continuing, it can give two 1-branches or one 2-branch of $\mathcal{C}$. Notice that in this case f_{n+1} is still in $\mathbb{C}[x,y]$, since $i_1 \equiv 0 \pmod 3$.

Case 4.2.3: If $c_1 = c_2 = c_3$, i.e., if $g_n^{\mathbf{a}}$ has one triple root, the algorithm $A_{c_1,\frac{i_1}{3}}$ goes on. Notice that in this case, f_{n+1} is still in $\mathbb{C}[x,y]$, since $i_1 \equiv 0 \pmod 3$.

Case 5.1: $\Gamma(f_n) = \{\mathbf{a}, \mathbf{b}\}, h(\mathbf{a}) = 2, h(\mathbf{b}) = 1$ and $i_1 \not\equiv 0 \pmod 2$, so we have

$$g_n^{\mathbf{a}} = {}_na_{03}y_n^2 + {}_na_{i_11}x^{i_1}, \qquad g_n^{\mathbf{b}} = {}_na_{i_11}y_n + {}_na_{i_20}x^{i_2-i_1}$$

$g_n^{\mathbf{a}}$ has two distinct roots of multiplicity 1: $c_1 x^{\frac{i_1}{2}}, c_2 x^{\frac{i_1}{2}}$, with $c_1^2 = -\frac{{}_n a_{i_1 1}}{{}_n a_{03}}$, $c_2 = -c_1$, which give equivalent parameterizations of the same 2-branch of $\mathcal{C}$, while $g_n^{\mathbf{b}}$ has one root of multiplicity 1: $-\frac{{}_n a_{i_2 0}}{{}_n a_{i_1 1}} x^{i_2 - i_1}$, giving one 1-branch. The algorithms stop here.

Case 5.2: $\Gamma(f_n) = \{\mathbf{a}, \mathbf{b}\}, h(\mathbf{a}) = 2, h(\mathbf{b}) = 1$ and $i_1 \equiv 0 \pmod 2$, so we have

$$g_n^{\mathbf{a}} = {}_n a_{03} y_n^2 + {}_n a_{\frac{i_1}{2} 2} x^{\frac{i_1}{2}} y_n + {}_n a_{i_1 1} x^{i_1}, \qquad g_n^{\mathbf{b}} = {}_n a_{i_1 1} y_n + {}_n a_{i_2 0} x^{i_2 - i_1}$$

$g_n^{\mathbf{b}}$ has one root of multiplicity 1, which gives a 1-branch of $\mathcal{C}$ and the relative algorithm stops here, while, if $c_1 x^{\frac{i_1}{2}}, c_2 x^{\frac{i_1}{2}}$ denote the roots of $g_n^{\mathbf{a}}$, we distinguish two cases (notice that ${}_n a_{\frac{i_1}{2} 2}$ may be 0, in which case, we are apparently in the same situation as Case 5.1):

Case 5.2.1: If $c_1 \neq c_2$, i.e., if $g_n^{\mathbf{a}}$ has two roots of multiplicity 1, each of them gives a 1-branch of $\mathcal{C}$, so we have three 1-branches of $\mathcal{C}$ and the algorithms stop here.

Case 5.2.2: If $c_1 = c_2$, i.e., if $g_n^{\mathbf{a}}$ has the double root $c_1 x^{\frac{i_1}{2}}$, the algorithm $A_{c_1, \frac{i_1}{2}}$ does not stop; continuing, it can give two 1-branches or one 2-branch of $\mathcal{C}$, hence, we may find three 1-branches or one 2-branch and one 1-branch, in all. Notice that in this case, f_{n+1} is still in $\mathbb{C}[x, y]$, since $i_1 \equiv 0 \pmod 2$.

Case 6.1: $\Gamma(f_n) = \{\mathbf{a}, \mathbf{b}\}, h(\mathbf{a}) = 1, h(\mathbf{b}) = 2$ and $i_1 + i_2 \not\equiv 0 \pmod 2$, so we have

$$g_n^{\mathbf{a}} = {}_n a_{03} y_n + {}_n a_{i_1 2} x^{i_1}, \qquad g_n^{\mathbf{b}} = {}_n a_{i_1 2} y_n^2 + {}_n a_{i_2 0} x^{i_2 - i_1}.$$

Since $i_2 - i_1 \not\equiv 0 \pmod 2$, the discussion is analogous to Case 5.1: $\mathcal{C}$ has one 1-branch and one 2-branch, and the algorithms stop here.

Case 6.2: $\Gamma(f_n) = \{\mathbf{a}, \mathbf{b}\}, h(\mathbf{a}) = 1, h(\mathbf{b}) = 2$ and $i_1 + i_2 \equiv 0 \pmod 2$, so we have

$$g_n^{\mathbf{a}} = {}_n a_{03} y_n + {}_n a_{i_1 2} x^{i_1}, \qquad g_n^{\mathbf{b}} = {}_n a_{i_1 2} y_n^2 + {}_n a_{\frac{i_1 + i_2}{2} 1} x^{\frac{i_2 - i_1}{2}} y_n + {}_n a_{i_2 0} x^{i_2 - i_1}.$$

Notice that the assumption $i_1 + i_2 \equiv 0 \pmod 2$ gives $i_2 - i_1 \equiv 0 \pmod 2$, hence, the discussion is analogous to Case 5.2, namely, we distinguish two cases:

Case 6.2.1: if $g_n^{\mathbf{b}}$ has two roots of multiplicity 1, each of them gives a 1-branch of $\mathcal{C}$, so we have globally three 1-branches of $\mathcal{C}$ and the algorithms stop here. Notice that ${}_n a_{\frac{i_1 + i_2}{2} 1}$ may be 0, in which case, we are apparently in the same situation as Case 6.1, but here $i_2 - i_1 \equiv 0 \pmod 2$.

Case 6.2.2: if $g_n^{\mathbf{b}}$ has one double root $c_1 x^{\frac{i_1}{2}}$, the relative algorithm does not stop; continuing, it can give two 1-branches or one 2-branch of $\mathcal{C}$, hence, globally, we may find three 1-branches or one 2-branch and one 1-branch. Notice that in this case, f_{n+1} is still in $\mathbb{C}[x, y]$, since $i_1 + i_2 \equiv 0 \pmod 2$.

Case 7: $\Gamma(f_n) = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}, h(\mathbf{a}) = h(\mathbf{b}) = h(\mathbf{c}) = 1$ and we have

$$g_n^{\mathbf{a}} = {}_n a_{03} y_n + {}_n a_{i_1 2} x^{i_1}, \qquad g_n^{\mathbf{b}} = {}_n a_{i_1 2} y_n + {}_n a_{i_2 1} x^{i_2 - i_1}, \qquad g_n^{\mathbf{c}} = {}_n a_{i_2 1} y_n + {}_n a_{i_3 0} x^{i_3 - i_2}$$

$g_n^{\mathbf{a}}$, $g_n^{\mathbf{b}}$ and $g_n^{\mathbf{c}}$ have one simple root each and the three roots give a 1-branch of $\mathcal{C}$ each: the algorithms stop here.

Remark 7. Notice that, in all the cases we have just considered, we obtain a polynomial f_{n+1} (for the following step of the Algorithm) which is always in $\mathbb{C}[x, y]$, i.e., we are always in case $n < \overline{n}$.

4.2. A Theorem for Triple Points

Now we are ready to see what happens when we run the Newton-Puiseux Algorithm 1 in the case of a triple point with a triple tangent.

Theorem 2. *Let* $\mathcal{C} : f(x,y) = 0$ *be a plane curve of degree d, such that O is a triple point for* $\mathcal{C}$ *and the x-axis is its triple tangent at O, so that*

$$f(x,y) = y^3 + \sum_{i+j=4}^{d} a_{ij}x^i y^j.$$

With the previous notation, the output of the Newton–Puiseux Algorithm 1 is as follows:

A finite sequence of n steps like case **4.2.3** *with* $n \geq 0$*, and then the following four cases are possible:*

(i) *step* $n+1$ *is a case* **4.1***; the algorithm stops:* $\mathcal{C}$ *has one 3-branch at O, and a parameterization for it is of the form*

$$\left(T^3, b_0T^{3h_0} + \cdots + b_nT^{3h_n} + c_1T^s + \ldots\right) \quad s \not\equiv 0 \ (\mathrm{mod}\ 3);$$

(ii) *step* $n+1$ *is one of the following cases:* **4.2.1**, **5.2.1**, **6.2.1**, **7***; the algorithm stops:*
$\mathcal{C}$ *has three 1-branches at O, each of which with a parameterization of the form*

$$\left(T, b_0T^{h_0} + \cdots + b_nT^{h_n} + c_jT^{q_j} \ldots\right)$$

where $q_1 = q_2 = q_3$ *in case* **4.2.1**, $q_1 = q_2$ *in cases* **5.2.1**, **6.2.1***;*

(iii) *step* $n+1$ *is one of the following cases:* **5.1**, **6.1***; the algorithm stops:*
$\mathcal{C}$ *has one 1-branch and one 2-branch at O, with parameterizations of the form:*

$$\left(T, b_0T^{h_0} + \cdots + b_nT^{h_n} + c_1T^q + \ldots\right)$$

$$\left(T^2, b_0T^{2h_0} + \cdots + b_nT^{2h_n} + d_1T^{2h+1} + \ldots\right).$$

(iv) *step* $n+1$ *is one of the following cases:* **4.2.2**, **5.2.2**, **6.2.2***; in order to decide if* $\mathcal{C}$ *has one 1-branch and one 2-branch or three 1-branches at O, the sub-algorithm relative to the double root must go on, so that we find a finite sequence of* $m \geq 0$ *steps like case* **2.2.2***, and then:*
if the next step is a case **2.1***, the algorithm stops:* $\mathcal{C}$ *has one 1-branch and one 2-branch at O, with parameterizations of the form*

$$\left(T, b_0T^{h_0} + \cdots + b_nT^{h_n} + c_1T^p + \ldots\right)$$

$$\left(T^2, b_0T^{2h_0} + \cdots + b_nT^{2h_n} + c_2T^{2p} + d_1T^{2k_1} + \cdots + d_mT^{2k_m} + a_1T^{2h+1} + \ldots\right)$$

if the next step is a case **2.2.1** *or* **3***, the algorithm stops:* $\mathcal{C}$ *has three 1-branches at O, with parameterizations of the form*

$$\left(T, b_0T^{h_0} + \cdots + b_nT^{h_n} + c_1T^p + \ldots\right)$$

$$\left(T, b_0T^{h_0} + \cdots + b_nT^{h_n} + c_2T^p + d_1T^{k_1} + \cdots + d_mT^{k_m} + e_1T^{q_1} + \ldots\right)$$

$$\left(T, b_0T^{h_0} + \cdots + b_nT^{h_n} + c_2T^p + d_1T^{k_1} + \cdots + d_mT^{k_m} + g_1T^{q_2} + \ldots\right)$$

where $q_1 = q_2$ *in case* **2.2.1**.

Proof. First, we claim that, in order to compute the number and multiplicities of the branches of $\mathcal{C}$ at a triple point with a triple tangent, it is enough to consider Newton polygons whose vertices are in $\mathbb{N}^2$; that is, applying the Newton–Puiseux Algorithm to f, we encounter only the cases described in Remark 6 above.

Proof of the claim. This is true for $n = 0$, since $f \in \mathbb{C}[x,y]$. Now, we choose a path as described in Algorithm 1, and assume the claim is true for the steps $1, \ldots, n$; then the

Newton Polygon of f_n is one of those described above. If the algorithm does not stop, we are in one of the following five cases: **2.2.2, 4.2.2, 4.2.3, 5.2.2, 6.2.2**, and since in each of these cases the vertices of the Newton Polygon of f_{n+1} are again in $\mathbb{N}^2$, the same is true for $n+1$. On the other hand, if the algorithm stops, we do not need to go any further, so the claim is true.

Now, we are going to analyze what happens in the assumption that $y_n \nmid f_n$; if, on the contrary, $y_n \mid f_n$, there is a branch coming from an algebraic component of $\mathcal{C}$, and thus, the corresponding series is in fact a polynomial.

Assume that after n steps, we are in case **4.2.3**. Since $3 \geq h(\Gamma(f)) \geq h(\Gamma(f_m)) \geq h(\Gamma(f_n)) = 3$ for $m \leq n$, we have $h(\Gamma(f_m)) = 3$ for $m \leq n$, hence, each previous step was one of the cases between **4.1** and **7**. If one of the previous steps was a case among **4.1, 4.2.1, 5.1, 5.2.1, 6.1, 6.2.1, 7**, we would have already stopped. If one of the previous steps was a case among **4.2.2, 5.2.2, 6.2.2**, only the sub-algorithm relative to the double root would have gone on, and in the next steps, the height of the Newton Polygon would have been 2. Therefore, all the previous steps were of type **4.2.3**.

Let us start with the step 0 which is a case **4.2.3**, and set $k = \frac{i_1}{3}$; since $i_1 \equiv 0 \pmod 3$, k is a positive integer, and for a suitable coefficient b, we have

$$f = (y - bx^k)^3 + \sum_{i+kj>3k} a_{ij}x^i y^j.$$

Applying the algorithm, we obtain: $g^{\mathbf{a}} = f^{\mathbf{a}} = (y - bx^k)^3$, $q_0 = bx^k$,

$$f(x, x^k(b+y_1)) = (x^k(b+y_1) - bx^k)^3 + \sum_{i+kj>3k} a_{ij}x^i(x^k(b+y_1))^j = x^{3k}y_1^3 + \sum_{i+kj>3k} a_{ij}x^{i+jk}(b+y_1)^j$$

and since each power of x has an exponent $\geq 3k$, we find

$$f_1 = \frac{x^{3k}y_1^3 + \sum_{i+kj>3k} a_{ij}x^{i+jk}(b+y_1)^j}{x^{3k}} = y_1^3 + \dots$$

If f_1 is a case **4.2.3**, again we have:

$$f_1 = (y - b_1x^h)^3 + \sum_{i+hj>3h} b_{ij}x^i y^j$$

and we find $q_1 = b_1x^{k_1}$ with $k_1 \in \mathbb{N}$; hence, if the sequence of cases **4.2.3** goes on to step n, we have a sequence of roots, setting $b_0 := b, k_0 := k$ (notice that $k_0 \geq 2$, otherwise, the tangent cone of C at O would be different from $y^3 = 0$)

$$q_0 = b_0x^{k_0}, \quad q_1 = b_1x^{k_1}, \quad \dots \quad q_n = b_nx^{k_n}, \qquad k_0, k_1, \dots, k_n \text{ positive integers.}$$

Let us assume that the algorithm does not stop and continutes with an infinite sequence of cases **4.2.3**; then, the Puiseux series

$$p(x) = \sum_{n \geq 0} b_n x^{k_0 + \dots + k_n}$$

is in fact a power series giving a unique 1-branch counted 3 times, but our curve $\mathcal{C}$ is reduced and this is not possible.

Hence, the only possible way in which **4.2.3** can appear is as follows: we have a finite sequence of n steps **4.2.3** and then one of the other cases.

If the step $n+1$ is a case **4.1**, $\mathcal{C}$ has one 3-branch at O, we have found:

$$q_0 = b_0x^{k_0},\ q_1 = b_1x^{k_1},\ \dots\ q_n = b_nx^{k_n},\ q_{n+1} = c_1x^{\frac{i_1}{3}}, \quad k_0, k_1, \dots, k_n \geq 0 \text{ integers}, i_1 \not\equiv 0 \pmod 3$$

which gives us

$$p(x) = \sum_{\ell=0}^{n} b_\ell x^{k_0+\ldots k_\ell} + c_1 x^{k_0+\cdots+k_n+\frac{i_1}{3}}$$

and since the common lowest denominator of $p(x)$ is 3, a parameterization for the 3-branch is

$$\left(T^3, b_0 T^{3k_0} + \cdots + b_n T^{3(k_0+\cdots+k_n)} + c_1 T^{3(k_0+\cdots+k_n)+i_1} + \ldots\right)$$

with $k_0, k_1, \ldots, k_n \geq 0$ integers, and $i_1 \not\equiv 0 \pmod 3$. Thus, we obtain (i).

If the step $n+1$ is not a case **4.1**, then it has to be one of the cases from **4.2.1** to **7** (except **4.2.3**), since the height of the Newton Polygon is still 3. It is easily seen that cases **4.2.1, 5.2.1, 6.2.1, 5.1, 6.1** give (ii) and (iii) just following the discussion in Remark 6.

If the step $n+1$ is one of the cases **4.2.2, 5.2.2, 6.2.2**, in order to decide if $\mathcal{C}$ has one 1-branch and one 2-branch or three 1-branches at O, the sub-algorithm relative to the double root must go on; now, the height of the Newton Polygon is decreased to 2, so that we have to use cases from **2.1** to **3**, and again, it is immediate to see that (iv) holds following the discussion in the previous Remark 6. □

We give two examples studied with the Algorithm as in the Theorem above.

Example 4. Let us consider the curve:

$$\mathcal{C} : y^3 - 2x^2y^2 + 4x^3y^2 + x^4y - x^4y^2 - 4x^5y + 3x^5y^2 + 6x^6y - 11x^7y - x^8 + 12x^8y + 7x^9 - 16x^{10} + 13x^{11} - 3x^{12} - x^{13} + y^{14} = 0.$$

We have that $m_{\mathcal{C}}(O) = 3$ and the unique tangent of $\mathcal{C}$ at O is $r : y = 0$. By applying Algorithm 1, one finds that $\mathcal{C}$ has at O one 2-branch given by $(T^2, T^4 - 2T^6 + T^7 + \ldots)$ and one 1-branch given by $(T, T^4 + \ldots)$.

Example 5. Let us consider the curve:

$$\mathcal{C} : y^3 - x^2y^2 + 2x^3y^2 - 2x^5y + x^5y^2 - x^7y + 2x^8y - 2x^{10} + x^{11} = 0.$$

We have that $m_{\mathcal{C}}(O) = 3$ and the unique tangent of $\mathcal{C}$ at O is $r : y = 0$. By applying the Algorithm 1 one finds that $\mathcal{C}$ has three 1-branches at O given by $(T, T^2 + \ldots)$, $(T, -T^3 + \ldots)$, and $(T, -T^5 + \ldots)$.

Our last observation is on*cuspidal* triple points, i.e., case (i) of the theorem above.

Definition 8. In case (i), we say that the triple point O is of *type s*. Notice that this definition is well-posed: if $(T^3, p(T))$, $(T^3, \tilde{p}(T))$ are two equivalent parameterization of the 3-branch, by Remark 1, we have the same value of s for the two parameterizations.

Remark 8. We point out this definition because in case (i), the type s is an invariant which determines the topological structure of the singularity; in fact, if a plane algebraic curve $\mathcal{C}$ has a triple point with one 3-branch at O, of type s, then it is locally topologically equivalent, see [19], to one of type $\left(T^3, T^s\right)$, i.e., with equation $y^3 - x^s$.

Notice that, in fact, a complete description of the analytical equivalence classes for this kind of triple points can be given, see ([page 4], [20]).

Example 6. Let us consider the following curve:

$$\mathcal{C} : y^3 - 3x^2y^2 + 3x^3y^2 + 3x^4y - 6x^5y - x^6 + 3x^6y + 3x^7 - 3x^8 + x^9 + x^{10} - y^{11} = 0.$$

We have that $m_{\mathcal{C}}(O) = 3$ and the unique tangent of $\mathcal{C}$ at O is $r : y = 0$. By applying Algorithm 1, one finds that $\mathcal{C}$ has one 3-branch at O given by $(T^3, T^6 - T^9 - T^{10} + \ldots)$;

thus, we are in case (i) of the theorem, and the triple point of $\mathcal{C}$ at O is of type 10, hence, it is topologically equivalent to $y^3 - x^{10}$.

Author Contributions: Conceptualization, S.C., A.G. and M.I.; Investigation, S.C., A.G. and M.I.; Writing—original draft, S.C., A.G. and M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

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

References

1. Bernardi, A.; Gimigliano, A.; Idà, M. Singularities of plane rational curves via projections. *J. Symb. Comput.* **2018**, *86*, 189–212. [CrossRef]
2. Chen, F.; Wang, W.; Liu, Y. Computing singular points of plane rational curves. *J. Symb. Comput.* **2008**, *43*, 92–117. [CrossRef]
3. Cox, D.; Kustin, A.R.; Polini, C.; Ulrich, B. A Study of singularities of rational curves via syzygies. *Mem. Amer. Math. Soc.* **2013**, *222*, 1045. [CrossRef]
4. Puiseux, V.A. Recherches sur les fonctions algébriques. *J. Math. Pures Appl.* **1850**, *15*, 365–480.
5. Newton, I. Letter from Newton to Oldenburg dated October 24, 1696. In *The Correspondence of Isaak Newton*; Cambridge University Press: Cambridge, UK,1960; Volume 2, pp. 110–161.
6. Newton, I. *The Method of Fluxions and Infinite Series: With Its Application to the Geometry of Curve-Lines*; John Nourse: London, UK, 1736.
7. van der Waerden, B.L. *Einführung in Die Alebraische Geometrie*; Springer: Berlin, Germany, 1939.
8. Abhyankar, S.S. *Algebraic Geometry for Scientists and Engineers*; Mathematical Surveys and Monographs 35; A.M.S.: Providence, RI, USA, 1990.
9. Walker, R.J. *Algebraic Curves*; Princeton University Press: Princeton, NJ, USA, 1950.
10. Arrondo, E. Apuntes de Curvas Algebraicas. 2020. Available online: http://www.mat.ucm.es/~arrondo/curvas.pdf (accessed on 3 October 2022).
11. Fischer, G. *Plane Algebraic Curves*; Kay, L., Translator; Student Math. Lib. AMS: Providence, RI, USA, 2001.
12. Greuel, G.M.; Lossen, C.; Shustin, E. *Introduction to Singularities and Deformations*; Springer: Berlin/Heidelberg, Germany, 2007.
13. Bassel, M.; Coquand, T. Dynamic Newton—Puiseux theorem. *J. Log. Anal.* **2013**, *5*, 1–22.
14. Poteaux, A.; Rybowicz, M. Complexity bounds for the rational Newton-Puiseux algorithm over finite fields. *AAECC* **2011**, *22*, 187–217. [CrossRef]
15. Poteaux, A.; Rybowicz, M. Good reduction of Puiseux series and applications. *J. Symb. Comput.* **2012**, *47*, 32–63. [CrossRef]
16. Duval, D. Rational Puiseux expansions. *Compos. Math.* **1989**, *70*, 119–154.
17. Manetti, M. *Corso Introduttivo alla Geometria Algebrica*; Scuola Normale Superiore di Pisa: Pisa, Italy, 1998. Available online: https://edizioni.sns.it/wp-content/uploads/2014/11/APPUNTI-MANETTI.pdf (accessed on 4 January 2023).
18. Poteaux, A.; Weimann, M. Computing Puiseux series: A fast divide and conquer algorithm. *Annales Henri Lebesgue* **2021**, *4*, 1061–1102. [CrossRef]
19. Oaku, T. Bernstein-Sato polynomials and analytic non-equivalence of plane curve singularities. *arXiv* **2022**, arXiv:2207.00699.
20. Hefez, A.; Hernandez, M.E. Classification of Plane Branches up to Multiplicity 4 . *J. Symb. Comput.* **2009**, *44*, 626–634. [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.