EFFECTIVE COMPUTATION OF BASE POINTS OF IDEALS IN TWO-DIMENSIONAL LOCAL RINGS

MARIA ALBERICH-CARRAMIÑANA, JOSEP ÀLVAREZ MONTANER, AND GUILLEM BLANCO

Abstract. We provide an algorithm that allows to describe the minimal log-resolution of an ideal in a smooth complex surface from the minimal log-resolution of its generators. In order to make this algorithm effective we present a modified version of the Newton-Puiseux algorithm that allows to compute the Puiseux decomposition of a product of not necessarily reduced or irreducible elements together with their algebraic multiplicity in each factor.

1. Introduction

Let (X, O) be a germ of smooth complex surface and $\mathcal{O}_{X,O}$ the ring of germs of holomorphic functions in a neighborhood of O, which we identify with $\mathbb{C}\{x, y\}$ by taking local coordinates. Let $\mathfrak{a} \subseteq \mathcal{O}_X$ be an ideal sheaf. From now on, if no confusion arises, we will indistinctly denote by $\mathfrak{a}$ the sheaf ideal or its stalk at O. In this later case we will be considering an ideal $\mathfrak{a} \subseteq \mathbb{C}\{x, y\}$. We also denote $\mathfrak{m} = \mathfrak{m}_{X,O} \subseteq \mathbb{C}\{x, y\}$ the maximal ideal.

A *log-resolution* of the pair $(X, \mathfrak{a})$, or a log-resolution of $\mathfrak{a}$ for short, is a proper birational morphism $\pi : X' \to X$ such that X' is smooth, the preimage of $\mathfrak{a}$ is locally principal, that is $\mathfrak{a} \cdot \mathcal{O}_{X'} = \mathcal{O}_{X'}(-F)$ for some effective Cartier divisor F, and $F + E$ is a divisor with simple normal crossings support where $E = Exc(\pi)$ is the exceptional locus. We point out that F decomposes into its affine and exceptional part $F = F_{\text{aff}} + F_{\text{exc}}$ according to its support. In particular, $F = F_{\text{exc}}$ when $\mathfrak{a}$ is $\mathfrak{m}$–primary.

In order to describe the divisor F we will use the theory of *weighted clusters* developed by Casas-Alvero in [7]. Namely, any log-resolution is a composition of blow-ups of points infinitely near to O. Hence, attached to F, there is a pair $\mathcal{K} = (K, v)$ where K is the set of infinitely near points that have been blown-up and $v : K \longrightarrow \mathbb{Z}$ is a valuation map that encodes the coefficients of the exceptional components in F. If E_i is the exceptional divisor that arises from the blowing-up of a point p_i, we have $F_{\text{exc}} = \sum_i d_i E_i$ where $v(p_i) = d_i$. The *weighted cluster of base points* $BP(\mathfrak{a}) = (B, \beta)$ of an ideal $\mathfrak{a}$ is the weighted cluster associated to the *minimal* log-resolution of $\mathfrak{a}$. Indeed, the version that we present in this work is a mild generalization of the construction given in [7] for the case of $\mathfrak{m}$–primary

Key words and phrases. Minimal log-resolution, weighted clusters, Newton-Puiseux algorithm.

All three authors are supported by Spanish Ministerio de Economía y Competitividad MTM2015-69135-P. MAC and JAM are also supported by Generalitat de Catalunya 2014SGR-634 project and they are with the Barcelona Graduate School of Mathematics (BGSMath). MAC is also with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC).

ideals (see Remark 2.2). Roughly speaking, it is the cluster of base points of $\mathfrak{a}$ weighted by the values of generic elements of $\mathfrak{a}$.

The aim of this work is to provide an algorithm that describes the minimal log-resolution of any ideal $\mathfrak{a}$ or equivalently, its associated weighted cluster of base points $BP(\mathfrak{a})$. If $\mathfrak{a} = (f)$ is a principal ideal with $f \in \mathbb{C}\{x, y\}$, the minimal log-resolution of $\mathfrak{a}$ equals the minimal log-resolution of the reduced curve ξ_{red} of $\xi : f = 0$. Indeed, computer algebra systems such as `Singular` [10] or `Magma` [6] can compute the divisor F whenever f is reduced.

If the ideal $\mathfrak{a} = (a_1, \dots, a_r) \subseteq \mathbb{C}\{x, y\}$ is not principal, the minimal log-resolution π of $\mathfrak{a}$ is no longer straightforwardly deduced from the minimal log-resolutions $\pi_i : X_i' \longrightarrow X$ of the principal ideals $\mathfrak{a}_i = (a_i)$ corresponding to each generator. Neither π dominates any π_i, nor the minimal proper birational morphism $\pi' : Y \longrightarrow X$ dominating all π_i, which is the minimal log-resolution of the principal ideal $(a_1 \cdots a_r)$, dominates π. Clearly, π, π' and any π_i factor through the blow-up of the *infinitely near points* that are common to all curves $a_i = 0$. Apart from this, no other inclusion between infinitely near points attached to $\mathfrak{a}$ and the principal ideals $\mathfrak{a}_i$ hold. In this work we will describe the minimal log-resolution of $\mathfrak{a}$ from the minimal log-resolution of its generators. We will not only provide the infinitely near points which must be blown-up and blown-down to reach π from those of π_i, but we will also describe the divisor F in terms of the divisors F_i, with $\mathfrak{a}_i \cdot \mathcal{O}_{X'} = \mathcal{O}_{X'}(-F_i)$.

The structure of the paper is as follows. In Section 2 we review the basics on the theory of weighted clusters. In particular, we introduce the weighted cluster of base points $BP(\mathfrak{a})$ associated to any ideal $\mathfrak{a} = (a_1, \dots, a_r) \subseteq \mathbb{C}\{x, y\}$. Our definition is a mild generalization of the weighted cluster defined by Casas-Alvero in [7, §7.2] for the case of $\mathfrak{m}$–primary ideals. Indeed, we have a decomposition $\mathfrak{a} = (g) \cdot \mathfrak{a}'$, where $g \in \mathbb{C}\{x, y\}$ is the greatest common divisor of the generators of $\mathfrak{a}$ and $\mathfrak{a}'$ is $\mathfrak{m}$–primary. From the weighted cluster $BP(\mathfrak{a}')$ and the cluster of *singular* points of the reduced germ η_{red} of $\eta : g = 0$ we can describe $BP(\mathfrak{a})$ (see Section 2.2).

In Section 3 we provide an alternative description of $BP(\mathfrak{a})$ that will be more useful for our purposes. The advantage is that the *virtual values* of the weighted cluster depend on the values of the generators of the ideal. In Section 4 we present the main result of this paper. Namely, we provide an algorithm (see Algorithm 4.6) that allows to compute $BP(\mathfrak{a})$ for any given ideal $\mathfrak{a}$. The idea behind our method is to give a first approximation of $BP(\mathfrak{a})$ by means of a weighted cluster associated to the product of the generators of the ideal $\mathfrak{a}'$ and g. Then, using some technical results developed in Section 4.1, we construct some intermediate weighted clusters that lead to the desired result.

In Section 5 we provide a generalization of the Newton-Puiseux algorithm that makes Algorithm 4.6 effective. The main feature of this version is that, given a set of elements $f_1, \dots, f_r \in \mathbb{C}\{x, y\}$ not necessarily reduced or irreducible, it computes the Puiseux decomposition of the product $f = f_1 \cdots f_r$, that is, the Puiseux series of f along with their

algebraic multiplicities in each of the factors $f_1, \dots, f_r$. Indeed, our method provides all the information needed to recover both the decomposition of each factor and the decomposition of the whole product at the same time. One of the key ingredients is to use the *square-free factorization* of f.

Finally, we would like to mention that a log-resolution of an ideal is a sort of principalization. An alternative approach to the problem of principalization was given by Cassou-Noguès and Veys in [8]. They describe an algorithm that transforms an ideal $\mathfrak{a} = (a_1, \dots, a_r) \in \mathbb{C}\{x, y\}$ into a principal one by means of *Newton maps*. At the i–th step of the algorithm, the Newton map is determined from the initial forms of the transformed generators at the $(i-1)$–th step. However, no connection is shown between the transformations of the principal ideals (a_i) defined by each generator and the transformation of the whole ideal $\mathfrak{a}$. In contrast, our approach provides such a relationship. Furthermore, we not only describe the divisor F from the divisors F_i, but our procedure also allows a *topological* generalization avoiding the requirement of explicit generators: given the equisingularity class of the generators a_i and the contacts between any pair of branches of different generators, we explicitly describe the equisingularity class of a generic element of $\mathfrak{a}$.

2. Preliminaries

Let X be a smooth complex surface and $\mathcal{O}_{X,O} \cong \mathbb{C}\{x, y\}$ the ring of germs of holomorphic functions in a neighborhood of a smooth point $O \in X$. Consider a sequence of blow-ups above O,

$$\pi : X' = X_{r+1} \to X_r \to \cdots \to X_1 = X,$$

with $X_{i+1} = \mathrm{Bl}_{p_i} X_i$ for a point $p_i \in X_i$ blowing-down to $O \in X$.

Let $\mathrm{Div}(X')$ be the group of integral divisors in X', i.e. divisors of the form $D = \sum_i d_i E_i$ where the E_i are pairwise different (non necessarily exceptional) prime divisors and $d_i \in \mathbb{Z}$. Among them, we will consider divisors in the lattice $\Lambda_\pi := \mathbb{Z}E_1 \oplus \cdots \oplus \mathbb{Z}E_r$ of exceptional divisors and we will simply refer them as divisors with *exceptional support*. We have two different basis of this $\mathbb{Z}$-module given by the *total transforms* and the *strict transforms* of the exceptional components. For simplicity, we will also denote the strict transforms by E_i and the total transforms by $\overline{E}_i$. In particular, any divisor $D_{exc} \in \Lambda_\pi$ can be presented in two different ways

$$D_{\mathrm{exc}} = \sum_{i=1}^{r} v_i E_i = \sum_{i=1}^{r} e_i \overline{E}_i,$$

where the weights v_i (resp. e_i) are the *values* (resp. *multiplicities*) of D_{exc}. In general, any divisor $D \in \mathrm{Div}(X')$ has a decomposition $D = D_{\mathrm{exc}} + D_{\mathrm{aff}}$ into its *exceptional* and *affine* part according to its support.

The relation between values and multiplicities is given by the combinatorics of the configuration of exceptional divisors. To such purpose we will use the theory of *infinitely*

near points. Roughly speaking, the exceptional divisors will be parameterized by the sequence of points $p_i \in E_i \subseteq X_{i+1}$ that are blown-up to achieve $\pi : X' \longrightarrow X$.

A point p infinitely near to the origin O is a point lying on the exceptional divisor of the composition of a finite sequence of blow-ups. The set $\mathcal{N}_O$ of points p infinitely near to O can be viewed as the disjoint union of the exceptional divisors appearing at successive blow-ups above O. The points in X will be called *proper* points in order to distinguish them from the infinitely near ones. The set $\mathcal{N}_O$ is endowed with a partial order relation $\leq$ defined by $p \leq q$ if and only if q is infinitely near to p. In this case we will say that p *precedes* q.

Given two points $p \leq q$ infinitely near to O, we say that q is *proximate* to p if and only if q belongs to the exceptional divisor E_p as proper or infinitely near point. We will denote this relation by $q \to p$. By construction, an infinitely near point q is proximate to just one or two points. In the former case we say that q is a *free point*, in the later it is a *satellite point*.

Let $p_1 = O$ and $p_2, \ldots, p_r$ be the infinitely near points that appear in the successive blow-ups composing $\pi : X' \longrightarrow X$. Let $D = \sum_{i=1}^r e_i \overline{E}_i = \sum_{i=1}^r v_i E_i$ be a divisor with exceptional support. Then, we have the following relation:

$$v_i = e_i + \sum_{p_i \to p_j} v_j. \tag{2.1}$$

Indeed, we can encode all the proximity relations in the *proximity matrix* $P = (p_{i,j})$ defined as:

$$p_{i,j} := \begin{cases} 1 & \text{if } j = i, \\ -1 & \text{if } p_i \to p_j, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the vectors $\mathbf{e} = (e_1, \ldots, e_n)$ and $\mathbf{v} = (v_1, \ldots, v_n)$ of multiplicities and values satisfy the base change formula $\mathbf{e}^\top = P \cdot \mathbf{v}^\top$.

Remark 2.1. We may also consider the *intersection matrix* $N = (n_{i,j})$ defined as:

$$n_{i,j} := \begin{cases} 1 & \text{if } E_i \cap E_j \neq \emptyset, \quad i \neq j, \\ 0 & \text{if } E_i \cap E_j = \emptyset, \quad i \neq j, \\ -r_i - 1 & \text{if } i = j. \end{cases}$$

where r_i is the number of points proximate to p_i. Then, $N = -P^\top P$.

2.1. **Weighted clusters of infinitely near points.** In order to describe divisors with exceptional support we will use the theory of weighted clusters of infinitely near points developed by Casas-Alvero in [7]. In the sequel, we will fix all the basic notions that we will use but we encourage the interested reader to take a look at [7] in order to get a deeper insight.

A *cluster* is a finite subset $K \subsetneq \mathcal{N}_O$ of infinitely near points to the origin O such that, if $p \in K$, then any preceding point $q < p$ also belongs to K. A *weighted cluster* $\mathcal{K} = (K, e)$

is a cluster K, called the *underlying cluster* of $\mathcal{K}$, together with a map $e : K \longrightarrow \mathbb{Z}$, where $e_p := e(p)$ is the *virtual multiplicity* of $\mathcal{K}$ at p. Alternatively, we may also weight a cluster by a system of *virtual values* $v_p := v(p)$ given by a map $v : K \longrightarrow \mathbb{Z}$. Both multiplicities and values are related recursively by

$$v_p = e_p + \sum_{p \to q} v_q, \tag{2.2}$$

and the reason for maintaining this apparent redundancy will become clear after Section 2.2. *Consistent clusters* are those weighted clusters with non-negative excesses, where the *excess* of $\mathcal{K}$ at p_i is

$$\rho_p = e_p - \sum_{q \to p} e_q.$$

We say that it is *strictly consistent* if, furthermore, $e_p > 0$ for all $p \in K$.

We define the sum of two weighted clusters $\mathcal{K} = (K, e)$ and $\mathcal{K}' = (K', e')$ weighted by multiplicities as $\mathcal{K} + \mathcal{K}' = (K \cup K', e + e')$ extending $e_p = 0$ for $p \notin K' \backslash K$ and $e'_p = 0$ for $p \notin K \backslash K'$.

Let $\pi_K : X' \longrightarrow X$ be the sequence of blow-ups centered at the points of the weighted cluster $\mathcal{K}$. Let $\xi : f = 0$ be a germ of plane curve defined by $f \in \mathcal{O}_{X,O}$. The *total transform* of ξ is the pull-back $\overline{\xi} := \pi_K^* \xi$ and can be written as[1]

$$\overline{\xi} = \widetilde{\xi} + \sum_{p \in K} e_p(f) \overline{E}_p = \widetilde{\xi} + \sum_{p \in K} v_p(f) E_p,$$

where $e_p(f)$ (resp. $v_p(f)$) is the multiplicity (resp. value) of ξ at p, and $\widetilde{\xi}$ is the *strict transform* that can be also realized as the closure of $\widetilde{\xi} := \pi^{-1}(\xi - \{O\})$. Neither $e_p(f)$ nor $v_p(f)$ depend on the equation f defining the curve ξ. The *virtual transform* with respect to the weighted cluster $\mathcal{K}$ is defined as

$$\check{\xi} := \widetilde{\xi} + \sum_{p \in K} (v_p(f) - v_p) E_p.$$

If $v_p(f) \geq v_p$, for all $p \in K$ then, we say that the curve ξ *goes virtually* through $\mathcal{K}$. It *goes sharply* through $\mathcal{K}$ when $v_p(f) = v_p$, for all $p \in K$ and ξ has no singular points, see Section 2.2, outside those in $\mathcal{K}$.

Let $\mathcal{K}_{\leq p}$ be the weighted subcluster consisting on all the points preceding a given point $p \in K$ with the same weights as $\mathcal{K}$. If we denote by $\check{\xi}_p$ the virtual transform with respect to $\mathcal{K}_{\leq p}$, then, for all $p \in K$, we have

$$e_p(\check{\xi}_p) = v_p(f) - \sum_{p \to q} v_q. \tag{2.3}$$

[1]We use the notation $\overline{E}_p$ and E_p to emphasize that the exceptional component corresponds to the point $p \in K$.

2.2. **The weighted cluster associated to the minimal log-resolution.** We have a correspondence between clusters of infinitely near points to the origin $O \in X$ and divisors with exceptional support in X', where $\pi : X' \to X$ is a sequence of blow-ups centered at the points of the cluster. Namely, we have a correspondence between a cluster $\mathcal{K} = (K, v)$ and a divisor $D_{\mathcal{K}} = \sum_{p \in K} v_p E_p \in \Lambda_\pi$.

In the case that $\pi : X' \to X$ is the minimal log-resolution of an ideal $\mathfrak{a} \subseteq \mathbb{C}\{x, y\}$ we are going to describe the weighted cluster associated to the exceptional part of the Cartier divisor F such that $\mathfrak{a} \cdot \mathcal{O}_{X'} = \mathcal{O}_{X'}(-F)$.

• **The case of reduced plane curves:** Consider a principal ideal generated by a reduced function $f \in \mathbb{C}\{x, y\}$. It defines a germ of curve $\xi : f = 0$ at O, whose *branches* are the germs defined by the irreducible factors of f.

The affine part F_{aff} of the divisor F corresponds to the strict transform $\widetilde{\xi}$ of the curve. The weighted cluster associated to the divisor F_{exc} will be denoted $\mathcal{S}(\xi)$ and coincides with the weighted cluster defined in [7, §3.8]. Namely, consider the set $\mathcal{N}_O(\xi)$ of points $p \in \mathcal{N}_O$ infinitely near to the origin O *lying on* ξ, i.e. those points such that $e_p(f) > 0$. Such a point is *simple* (resp. *multiple*) if $e_p(f) = 1$ (resp. $e_p(f) > 1$). It is *singular* if it is either multiple, satellite or precedes a satellite point lying on ξ. Then, $\mathcal{S}(\xi)$ is the weighted cluster of singular points, weighted by the multiplicities, or the values, of ξ. It is a strictly consistent cluster since it satisfies the *proximity equalities* (see [7, 3.5.3]).

$$e_p(f) = \sum_{q \to p} e_q(f).$$

• **The case of ideals:** Given an ideal $\mathfrak{a} = (a_1, \ldots, a_r) \subseteq \mathbb{C}\{x, y\}$ we have a decomposition $\mathfrak{a} = (g) \cdot \mathfrak{a}'$, where $g \in \mathbb{C}\{x, y\}$ is the greatest common divisor of the generators of $\mathfrak{a}$ and $\mathfrak{a}' = (f_1, \ldots, f_r)$ is $\mathfrak{m}$–primary.

In the case that $\mathfrak{a}$ is indeed $\mathfrak{m}$–primary we have that the divisor $F = F_{\text{exc}}$ only has exceptional support. Its associated weighted cluster is the weighted cluster of *base points* $BP(\mathfrak{a})$ defined in [7, §7.2]. It consists of the points shared by the curves defined by generic elements of $\mathfrak{a}$ weighted by the corresponding multiplicities or values.

In the general case, we have a decomposition $F = F_{\text{aff}} + F_{\text{exc}}$, so we will have to treat the affine and the exceptional part of the divisor F separately. To describe the affine part, consider a decomposition of g into its irreducible factors $g = g_1^{a_1} \cdots g_t^{a_t}$. Then, we have $F_{\text{aff}} = a_1 E_1 + \cdots + a_t E_t$, where E_i is the strict transform on X' of the irreducible germ $\eta_i : g_i = 0$. In order to describe the weighted cluster associated to the exceptional part F_{exc} we will give the following natural generalization of the cluster of base points considered in [7, §7.2]:

Define the multiplicity of the ideal at O as $e_O(\mathfrak{a}) = e = \min\{e_O(f) \,|\, f \in \mathfrak{a}\}$. If $p \in E_O$ is any point in the first neighborhood of O, the pull-back of functions induces an injective homomorphism of rings $\varphi_p : \mathcal{O}_{X,O} \longrightarrow \mathcal{O}_{X_p,p}$. We consider the ideal defined as $\mathfrak{a}_p = z^{-e}\varphi_p(\mathfrak{a})$ (where z is any equation for the exceptional divisor E_O near p), that

is, generated by the germs $\check{\xi}_p : z^{-e}\varphi_p(f) = 0$, i.e. the virtual transforms of $\xi : f = 0$, $f \in \mathfrak{a}$. These definitions can be extended to any $p \in \mathcal{N}_O$, as well as the multiplicity $e_p(\mathfrak{a}) = e_p(\mathfrak{a}_p)$. Since the germs $\xi : f = 0$, $f \in \mathfrak{a}$, have no fixed part outside $\eta : g = 0$, the set of points such that $0 < e_p(\mathfrak{a}) \neq e_p(g)$ is finite, and hence it is a cluster. We then add to this weighted cluster the weighted cluster $\mathcal{S}(\eta_{\text{red}})$ of singular points of the reduced germ η_{red}, (weighted by the multiplicities of the ideal $\mathfrak{a}$ in those points) giving a larger cluster that we also refer as the *weighted cluster of base points* of $\mathfrak{a}$ and denote by $BP(\mathfrak{a})$. Observe that $BP(\mathfrak{a})$ does not depend on the decomposition $\mathfrak{a} = (g) \cdot \mathfrak{a}'$, namely on the unit affecting either g or the generators of $\mathfrak{a}'$.

Remark 2.2. Let $BP(\mathfrak{a}') = (K', e')$ be the cluster of base points of the $\mathfrak{m}$–primary ideal $\mathfrak{a}'$ and let $\mathcal{S}(\eta_{\text{red}}) = (S, e'')$ be the cluster of singular points of the reduced germ associated to $\eta : g = 0$. Then, the cluster $BP(\mathfrak{a}) = (B, e)$ can be also described as follows:

- $B = K' \cup S$.
- $e_p = e'_p + e_p(g)$, extending $e'_p = 0$ for p outside K'.

In particular, we have to consider the multiplicities $e_p(g)$ of the non-reduced germ η instead of the multiplicities e''_p of η_{red}. In fact, $BP(\mathfrak{a}) = BP((g)) + BP(\mathfrak{a}')$. If we weight $BP(\mathfrak{a}) = (B, \beta)$ and $BP(\mathfrak{a}') = (K', v')$ using values, then we have $\beta_p = v'_p + v_p(g)$.

We point out that, in the case of $\mathfrak{m}$–primary ideals, this description coincides with the one given in [7, §7.2]. Hence, we recover easily the properties in loc.cit. also in this case. Namely, $BP(\mathfrak{a})$ is strictly consistent and all germs $\xi : f = 0$, $f \in \mathfrak{a}$, go virtually through $BP(\mathfrak{a})$. Generic germs go sharply through it, they miss any fixed finite set of points not in $BP(\mathfrak{a}) \cup \mathcal{S}(\eta_{\text{red}})$, in particular have the same equisingularity class, and their fixed part reduces to g, having no multiple factors outside those of g. Furthermore, the ideal $\mathfrak{a}$ may be generated by a finite system of generators defining germs going sharply through $BP(\mathfrak{a})$.

3. An alternative characterization of the weighted cluster of base points of an ideal

The aim of this section is to give an alternative characterization of the weighted cluster of base points $BP(\mathfrak{a})$ associated to any ideal $\mathfrak{a} \subseteq \mathbb{C}\{x, y\}$. This new approach will be more suitable for our purposes in the rest of this work. For the sake of simplicity in the notations, we will assume that $\mathfrak{a} = (f_1, \ldots, f_r)$ is $\mathfrak{m}$–primary (see Remark 2.2). Then, we will describe a weighted cluster $\mathcal{K} = (K, v)$ with virtual values v depending on the values of the curves $\xi_i : f_i = 0, i = 1, \ldots, r$ and we will then prove that it is equal to $BP(\mathfrak{a})$.

Definition 3.1. Let $\mathfrak{a} = (f_1, \ldots, f_r) \subseteq \mathbb{C}\{x, y\}$ be an $\mathfrak{m}$–primary ideal. For any point $p \in \mathcal{N}_O$ equal or infinitely near to O we define the value $v_p := \min\{v_p(f_1), \ldots, v_p(f_r)\}$, and recursively on the proximate points, we define $h_O = 0$ and

$$h_p := \sum_{p \to q} v_q.$$

Define also the weighted cluster $\mathcal{K} = (K, v)$, where K is the set of points $p \in \mathcal{N}_O$ such that $h_p < v_p$. The corresponding virtual multiplicities are defined as $e_p = v_p - h_p$, for each $p \in K$.

To be a proper weighted cluster, we need to check that K is finite and that for any $p \in K$, all the preceding points also belong to K. To do so, we will start with a technical lemma.

Lemma 3.2. *Let $\mathfrak{a} = (f_1, \dots, f_r) \subseteq \mathbb{C}\{x, y\}$ be an $\mathfrak{m}$–primary ideal. Then, for any $f \in \mathfrak{a}$ and any $p \in \mathcal{N}_O$, we have $v_p(f) \geq \min\{v_p(f_1), \dots, v_p(f_r)\}$.*

Proof. Assume that $f = g_1 f_1 + \cdots + g_r f_r$, for $g_1, \dots, g_r \in \mathbb{C}\{x, y\}$. Using the fact that $v_p(\cdot)$ is a discrete valuation in $\mathbb{C}\{x, y\}$ (see [7, §4.5]), we have:

$$\begin{aligned} v_p(f) &= v_p(g_1 f_1 + \cdots + g_r f_r) \geq \min\{v_p(g_1 f_1), \dots, v_p(g_r f_r)\} \\ &= \min_i\{v_p(g_i) + v_p(f_i)\} \geq \min\{v_p(f_1), \dots, v_p(f_r)\}, \end{aligned}$$

where in the last inequality we used that $v_p(g) \geq 0, \forall g \in \mathbb{C}\{x, y\}$. □

Next, we prove that the definition of the weighted cluster $\mathcal{K} = (K, v)$ does not depend on the generators of the ideal.

Lemma 3.3. *Let $\mathfrak{a} = (f_1, \dots, f_r) \subseteq \mathbb{C}\{x, y\}$ be an $\mathfrak{m}$–primary ideal. The virtual values of the weighted cluster $\mathcal{K} = (K, v)$ associated to $\mathfrak{a}$ do not depend on the generators of the ideal. In other words, $v_p = \min_{f \in \mathfrak{a}}\{v_p(f)\}$.*

Proof. It is clear that $v_p = \min_i\{v_p(f_i)\} \geq \min_{f \in \mathfrak{a}}\{v_p(f)\}$ since $\{f_1, \dots, f_r\} \subset \mathfrak{a}$. On the other hand, by Lemma 3.2, $v_p(f) \geq \min_i\{v_p(f_i)\}$, for all $f \in \mathfrak{a}$, hence $\min_{f \in \mathfrak{a}}\{v_p(f)\} \geq v_p$ and the result follows. □

Lemma 3.4. *Under the assumptions of Definition 3.1, the inequality $h_p \leq v_p$ holds for any point $p \in \mathcal{N}_O$ equal or infinitely near to O.*

Proof. The inequality is clear when $p = O$. Now, assume that p is free, so it is proximate to one point $p \to q$. Then, we have:

$$h_p = v_q = \min_i\{v_q(f_i)\} \leq \min_i\{v_p(f_i)\} = v_p.$$

If p is satellite, it is proximate to two points $p \to q$ and $p \to q'$. Then, we have:

$$\begin{aligned} h_p = v_q + v_{q'} &= \min_i\{v_q(f_i)\} + \min_i\{v_{q'}(f_i)\} \\ &\leq \min_i\{v_q(f_i) + v_{q'}(f_i)\} = \min_i\{v_p(f_i)\} = v_p. \end{aligned}$$

□

Lemma 3.5. *Let $\mathfrak{a} = (f_1, \dots, f_r) \subseteq \mathcal{O}_{X,O}$ be an $\mathfrak{m}$–primary ideal. If there exists a generator f_i such that $h_p = v_p = v_p(f_i)$, then we have $e_p(f_i) = 0$ and $v_q = v_q(f_i)$ for any point $q \in \mathcal{N}_O$ such that $p \to q$.*

Proof. If p is a free point, we take the unique point q such that $p \to q$. Notice that

$$v_p(f_i) = v_p = h_p = v_q = \min_j\{v_q(f_j)\},$$

so we have $v_p(f_i) \leq v_q(f_i)$. It follows from Equation 2.2 that $v_p(f_i) = v_q(f_i)$ and $e_p(f_i) = 0$, hence, $v_q = v_q(f_i)$.

If p is satellite, we take the points q and q' such that $p \to q$, $p \to q'$. We have

$$v_p(f_i) = v_p = h_p = v_q + v_{q'},$$

thus $v_p(f_i) \leq v_q(f_i) + v_{q'}(f_i)$. Using Equation 2.2, we obtain $v_p(f_i) = v_q(f_i) + v_{q'}(f_i)$ and $e_p(f_i) = 0$. Finally, if $v_q < v_q(f_i)$ or $v_{q'} < v_{q'}(f_i)$, then $h_p = v_q + v_{q'} < v_p(f_i)$, so we get a contradiction. □

Proposition 3.6. *Under the assumptions of Definition 3.1, if $p \in K$, then any point q preceding p also belongs to K.*

Proof. We will prove the converse statement: assume that $q \notin K$, i.e. $h_q = v_q$. We will prove $h_p = v_p$ for any p in the first neighborhood of q, and it will follow inductively $h_p = v_p$, i.e. $p \notin K$, for any point p infinitely near to q.

Assume that $q \notin K$ and let p be a point in the first neighborhood of q, in particular $p \to q$. Consider a generator f_i such that $v_q = \min_j\{v_q(f_j)\} = v_q(f_i)$, hence $h_q = v_q = v_q(f_i)$. If p is satellite, we take the second point q' such that $p \to q'$. Then, by Lemma 3.5

$$h_p = v_q + v_{q'} = v_q(f_i) + v_{q'}(f_i) = v_p(f_i),$$

and by Lemma 3.4, $h_p = v_p(f_i) = v_p$. If p is free, the same reasoning is valid by taking $v_{q'} = v_{q'}(f_i) = 0$. □

We will show next that $\mathcal{K} = (K, v)$ equals the weighted cluster $BP(\mathfrak{a}) = (B, \beta)$ of base points of $\mathfrak{a}$ and we will conclude that K is finite.

Proposition 3.7. *Let $BP(\mathfrak{a}) = (B, \beta)$ be the weighted cluster of base points of a $\mathfrak{m}$–primary ideal $\mathfrak{a} = (f_1, \ldots, f_r) \subseteq \mathbb{C}\{x, y\}$ and $\mathcal{K} = (K, v)$ as given in Definition 3.1. Let $p \in B$, then $p \in K$ and the equality of virtual values $\beta_p = v_p$ is satisfied.*

Proof. Let $f = g_1 f_1 + \cdots + g_r f_r$ be the equation of a germ $\xi : f = 0$ going sharply through $BP(\mathfrak{a})$. From Lemma 3.2 we obtain

$$\beta_p = \ v_p(f) \geq \min_i\{v_p(f_i)\} = v_p.$$

Now, by [7, 7.2.16], we may find a system of generators $\mathfrak{a} = (h_1, \ldots, h_s)$ such that $\zeta_i : h_i = 0$ goes sharply through $BP(\mathfrak{a})$. Then,

$$v_p = \min_i\{v_p(f_i)\} \geq \min_i\{v_p(h_i)\} = \beta_p,$$

after applying Lemma 3.2, once again, to the elements f_i expressed in terms of $h_1, \ldots, h_s$. Therefore, the equality $v_p = \beta_p$ follows.

Since the same equality holds for all the points preceding p, we infer

$$v_p - h_p = \beta_p - \sum_{p \to q} \beta_q = b_p > 0,$$

i.e. $v_p > h_p$, since b_p is the virtual multiplicity at p of the strictly consistent weighted cluster $BP(\mathfrak{a})$. □

Theorem 3.8. *The weighted clusters $BP(\mathfrak{a}) = (B, \beta)$ and $\mathcal{K} = (K, v)$ are equal. In particular, K is finite.*

Proof. From Proposition 3.7 we already have $B \subseteq K$. We will prove the other inclusion using induction on the order of neighborhood which a point $p \in K$ belongs to.

For $p = O$, it is clear that p belongs to both K and B. Now, assume that the assertion is true for all the points preceding p, which are in K by Proposition 3.6. Let $q \in B$ be the antecessor of p. By [7, 7.2.6], $p \in B$ if and only if $0 < \min_{f\in\mathfrak{a}}\{e_p(\check{\xi}_p)\}$, where $e_p(\check{\xi}_p)$ is the virtual multiplicity of the germ $\xi : f = 0$ at p relative to the weighted cluster $BP(\mathfrak{a})_{\leq p}$. This is equivalent, by Equation 2.3, to

$$\min_{f\in\mathfrak{a}}\{v_p(f)\} > \sum_{p\to s} \beta_s.$$

By Lemma 3.3, $v_p = \min_{f\in\mathfrak{a}}\{v_p(f)\}$. Thus, applying Proposition 3.7 to the points preceding p, we have that p belongs to B if and only if

$$v_p > \sum_{p\to s} \beta_s = \sum_{p\to s} v_s = h_p.$$

□

Remark 3.9. Theorem 3.8 is a generalization for $\mathfrak{m}$–primary ideals of [1, 2.5] which describes the base points of a pencil $\lambda_1 f_1 + \lambda_2 f_2$ of curves $f_1, f_2 \in \mathbb{C}\{x, y\}$, $\lambda_1, \lambda_2 \in \mathbb{C}$.

Corollary 3.10. *Let $BP(\mathfrak{a}) = (B, \beta)$ be the weighted cluster of base points of a $\mathfrak{m}$–primary ideal $\mathfrak{a} = (f_1, \ldots, f_r) \subseteq \mathbb{C}\{x, y\}$. Any cluster of infinitely near points K' weighted by the values*

$$v(p) = v_p = \min_i\{v_p(f_i)\}, \quad \forall p \in K',$$

or, alternatively, weighted by the multiplicities

$$e(p) = e_p = v_p - \sum_{p\to q} v_q, \quad \forall p \in K',$$

satisfying $e_p \neq 0$ for any $p \in K'$ is a subcluster of B.

Proof. Since, by definition, $K = \{p \in \mathcal{N}_O \mid e_p > 0\}$, clearly $K' \subseteq K$. Then, the result follows using Theorem 3.8. □

4. An algorithm to compute the base points of an ideal

In this section we will provide an algorithm that allows to compute the weighted cluster of base points of any ideal $\mathfrak{a} = (a_1, \ldots, a_r) \subseteq \mathbb{C}\{x, y\}$. First we recall that we have a decomposition $\mathfrak{a} = (g) \cdot \mathfrak{a}'$, where $g \in \mathbb{C}\{x, y\}$ is the greatest common divisor of the generators of $\mathfrak{a}$ and $\mathfrak{a}' = (f_1, \ldots, f_r)$ is $\mathfrak{m}$–primary. Moreover, the weighted cluster of base points of $\mathfrak{a}$ is described in terms of the base points of $\mathfrak{a}'$ and the cluster of singular points of the reduced germ η_{red} of $\eta : g = 0$ (see Remark 2.2).

The cluster $\mathcal{S}(\eta_{\text{red}})$ is easy to describe using the Newton-Puiseux algorithm (see [7]) so the bulk of the process is in the computation of $BP(\mathfrak{a}')$. Before proceeding to describe the algorithm we need to introduce several technical results that will allow us to compute the weighted cluster $BP(\mathfrak{a}')$ in terms of the weighted cluster of singular points of the germs defined by the set of generators $\xi_i : f_i = 0, i = 1, \dots, r$.

4.1. **Adding free and satellite points.** For the sake of simplicity, we will assume throughout this subsection that our ideal $\mathfrak{a} = (f_1, \dots, f_r) \subseteq \mathbb{C}\{x, y\}$ is $\mathfrak{m}$–primary. In order to compute the weighted cluster $BP(\mathfrak{a}) = (B, \beta)$ we will start with a weighted cluster associated to the product of the generators $\xi : f_1 \cdots f_r = 0$ which gives a first approximation. Then, using the results of this section, we will add the necessary free and satellite points to this weighted cluster to obtain all the base points.

Assumption 4.1. Assume $\mathcal{K}' = (K', v)$ is a weighted cluster with a system of values $v_p = \min_i\{v_p(f_i)\}$ for any $p \in K'$ and satisfying that any point $p \in B$ singular for ξ_{red} is already in K'.

The following set of technical results will allow us to decide which points we need to add to K' in order to obtain all the base points. The first result states that all the free points in $BP(\mathfrak{a})$ lie on the generators.

Lemma 4.2. *Let q be a free point that does not lie on any curve $\xi_i : f_i = 0,\ \ i = 1, \dots, r$ then, $q \notin BP(\mathfrak{a})$.*

Proof. Let $q \to p$. If $q \notin \xi_i : f_i = 0$, for all i, then $v_q(f_i) = v_p(f_i)$ for all $i = 1, \dots, r$ and $v_q = \min_i\{v_q(f_i)\} = \min_i\{v_p(f_i)\} = v_p$, hence $e_q = v_q - v_p = 0$ and $q \notin BP(\mathfrak{a})$. □

The next result characterizes the free points in $BP(\mathfrak{a})$ that are not singular for the reduced germ ξ_{red} associated to the generators.

Proposition 4.3. *Let $\mathcal{K}'$ be a weighted cluster as in Assumption 4.1. Let $q \notin K'$ be a free point proximate to $p \in K'$. Then, q is in $BP(\mathfrak{a})$ if and only if any generator f_i with $v_p(f_i) = v_p$ satisfies $e_q(f_i) > 0$.*

Proof. We shall apply Theorem 3.8 to characterize whether q belongs to $BP(\mathfrak{a})$. By definition, $v_q = \min_j\{v_q(f_j)\}$ and $v_q(f_j) = e_q(f_j) + v_p(f_j)$, for $j = 1, \dots, r$.

Set $\Lambda := \{j \mid e_q(f_j) > 0\}$. Comparing

$$v_q = \min_{j\in\Lambda, k\notin\Lambda}\{v_p(f_k), v_p(f_j) + e_q(f_j)\} \quad \text{and} \quad v_p = \min_i\{v_p(f_i)\},$$

we infer $e_q = v_q - v_p > 0$ if and only if $v_p(f_k) > v_p$, $\forall k \notin \Lambda$ which is equivalent to $\{i \mid v_p(f_i) = v_p\} \subseteq \Lambda$. □

Remark 4.4. Under the hypothesis of Proposition 4.3 we observe that there might be two generators, say f_i, f_j, such that $e_q(f_i) > 0$, $e_q(f_j) > 0$ although q is not singular for the reduced germ of $f_1 \cdots f_r = 0$. This may happen when f_i and f_j have a common factor which is not a common factor of the rest of generators. This is a subtle difference with respect to the case of pencils treated in [1].

Our next result deals with the satellite base points not already in K'. Notice that these missing satellite points will not lie on any generator, otherwise they would belong to the singular points of ξ_{red}.

Proposition 4.5. *Let $\mathcal{K}'$ be a weighted cluster as in Assumption 4.1. Let $q \notin K'$ be a satellite point proximate to $p, p' \in K'$. Then, q is in $BP(\mathfrak{a})$ if and only if for each generator f_i either $v_p(f_i) > v_p$ or $v_{p'}(f_i) > v_{p'}$.*

Proof. Let us start by proving the converse implication. We know that $q \notin f_j$ for any $j = 1, \dots, r$, otherwise q would be in K'. We want to see that $e_q = v_q - v_p - v_{p'} > 0$. Then, $v_q = \min_j\{v_q(f_j)\} = \min_j\{v_p(f_j) + v_{p'}(f_j)\}$ and the last equality is true because $e_q(f_j) = 0$. By hypothesis, $v_p(f_j) + v_{p'}(f_j) > v_p + v_{p'}$, for any j, hence $v_q > v_p + v_{p'}$ as we wanted.

For the other implication, let us assume the contrary, that is, there exists a generator f_i such that $v_p(f_i) = v_p$ and $v_{p'}(f_i) = v_{p'}$. We know that $q \notin f_i$, otherwise it would be in K'. By definition, $v_q = \min_j\{v_q(f_j)\} = \min_j\{v_p(f_j) + v_{p'}(f_j)\} = v_p(f_i) + v_{p'}(f_i) = v_p + v_{p'}$, implying that $e_q = 0$, which is a contradiction with the fact that $q \in BP(\mathfrak{a})$. □

4.2. **An algorithm to compute the base points of an ideal.** In this subsection we go back to our original setup, so let $\mathfrak{a} = (a_1, \dots, a_r) \subseteq \mathbb{C}\{x, y\}$ be an ideal that admits a decomposition $\mathfrak{a} = (g) \cdot \mathfrak{a}'$, where $g \in \mathbb{C}\{x, y\}$ is the greatest common divisor of the generators of $\mathfrak{a}$ and $\mathfrak{a}' = (f_1, \dots, f_r)$ is $\mathfrak{m}$–primary. With all the technical results stated above and the relation between $BP(\mathfrak{a})$, $BP(\mathfrak{a}')$ and the cluster $\mathcal{S}(\eta_{\mathrm{red}})$ of singular points of $\eta : g = 0$ (see Remark 2.2), we present our algorithm.

Algorithm 4.6. *(Base points of an ideal)*

`Input:` *An ideal $\mathfrak{a} = (a_1, \dots, a_r) \subseteq \mathbb{C}\{x, y\}$.*
`Output:` *The weighted cluster of base points $BP(\mathfrak{a})$.*

i) *Find $g = \gcd(a_1, \dots, a_r)$ and set $a_i = g f_i$. Compute $f = g f_1 \cdots f_r$.*

ii) *Find the cluster $\overline{K}$ of singular points of the reduced germ ξ_{red}, where $\xi : f = 0$, and the system of virtual values $\{v_p(f_i)\}_{p \in \overline{K}}$, and $\{v_p(g)\}_{p \in \overline{K}}$.*

Compute $v_p = \min_i\{v_p(f_i)\}$ for $p \in \overline{K}$ and set the weighted cluster $\overline{\mathcal{K}} = (\overline{K}, v)$.

iii) *Define $\mathcal{K}' = (K', v)$ from $\overline{\mathcal{K}}$ by adding, if necessary, the missing free points using Proposition 4.3 weighted by the values $v_p = \min_i\{v_p(f_i)\}$ for each new $p \in K' \setminus \overline{K}$.*

iv) *Define $\mathcal{K}'' = (K'', v)$ from $\mathcal{K}'$ by adding, if necessary, the missing satellite points using Proposition 4.5, weighted by the values $v_p = \min_i\{v_p(f_i)\}$ for each new $p \in K'' \setminus K'$.*

v) *Compute, recursively on the order of neighborhood p belongs to, the multiplicities $e_p = v_p - \sum_{p \to q} v_q$.*

Define $\mathcal{K} = (K, v)$ with $K \subset K''$ containing the points $p \in \mathcal{K}$ such that $e_p \neq 0$ and the virtual values v_p.

vi) *Set* $BP(\mathfrak{a}') = \mathcal{K}$, *the weighted cluster of base points of* $\mathfrak{a}' = (f_1, \dots, f_r)$.

vii) *From* $\overline{K}$ *extract the cluster of singular points* $\mathcal{S}(\eta_{\text{red}}) = (S, v'')$, *where* $\eta : g = 0$.

viii) *Return* $BP(\mathfrak{a}) = (B, \beta)$, *where* $B = K \cup S$ *and* $\beta_p = v_p + v_p(g)$, $\forall p \in B$.

Our next result proves the correctness of the Algorithm.

Theorem 4.7. *Algorithm 4.6 computes* $BP(\mathfrak{a}) = (B, \beta)$, *the weighted cluster of base points of the ideal* $\mathfrak{a}$.

Proof. Since the cluster $\overline{K}$ fulfills the hypothesis of Proposition 4.3 and Proposition 4.5, we can use them to add the remaining base points.

After step iv) all the base points have been added. Indeed, if we had to add a missing base point in the first neighborhood of a point already in K', it would have to be free as we have added all the missing satellites in the last step. This free point would have to lie on a generator, by Lemma 4.2, and it would have to be after one of the new satellite points, otherwise we would have added it in the fourth step. But that is impossible because the new satellite points cannot lie on a generator, by Assumption 4.1, and hence, neither can do any of its successors.

By Corollary 3.10, after removing the points p in K'' such that $e_p = 0$, the resulting cluster K is inside B and since no base point is missing it must be equal to B. □

It is a classical result that the equisingularity class, and equivalently the topological class [13], [5], of any element $f = f_1^{\alpha_1} \cdots f_r^{\alpha_r} \in \mathcal{O}_{X,O}$ with $f_1, \dots, f_r$ irreducible, is determined by the equisingularity class of each component $f_i^{\alpha_i}$ together with the intersection multiplicities[2] $[f_i \cdot f_j]_O$, for all $i \neq j$. This can be described by means of the proximity matrix associated to the log-resolution of (f) and its vector of multiplicities. As a corollary of Theorem 4.7 we obtain the following generalization.

Corollary 4.8. *Given an ideal* $\mathfrak{a} = (a_1, \dots, a_r) \subseteq \mathcal{O}_{X,O}$, *the equisingular class of a generic element of* $\mathfrak{a}$ *is determined by the equisingularity class of each generator* a_i, *and the intersection multiplicities of every pair of branches from different generators.*

Proof. Generic elements in $\mathfrak{a}$ go sharply through the weighted cluster of base points $BP(\mathfrak{a})$. By Theorem 4.7, the relative position of the infinitely near points in $BP(\mathfrak{a})$ and the multiplicities, or values, are completely determined by the equisingularity class of each a_i and the intersection multiplicities between any pair of branches of different generators. □

Example 4.9. Consider the ideal

$$\mathfrak{a} = (a_1, a_2, a_3) = \left((y^5 + x^7)^2 + y^{10}x, x^8(y^3 + x^5), y^8(y^2 - x^3)\right) \subseteq \mathbb{C}\{x, y\}.$$

The steps of Algorithm 4.6 are performed as follows:

i) We have that $g = \gcd(a_1, a_2, a_3) = 1$, so the ideal is $\mathfrak{m}$–primary. Then, we compute the product of the generators $f = a_1 a_2 a_3$.

[2]Noether's intersection formula states that $[f \cdot g]_O = \sum_{p \in K} e_p(f) e_p(g)$.

ii) The cluster $\overline{K}$ of singular points of ξ_{red} with $\xi : f = 0$ is described by means of the proximity matrix:

$$P_{\overline{K}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

The virtual values $\{v_p(a_i)\}_{p\in\overline{K}}, i = 1, 2, 3$ are the following:

$$v(a_1) = [10\ 10\ 14\ 14\ 28\ 40\ 70\ 72\ 74\ 75\ 150\ 151\ 28\ 42\ 42]^T,$$

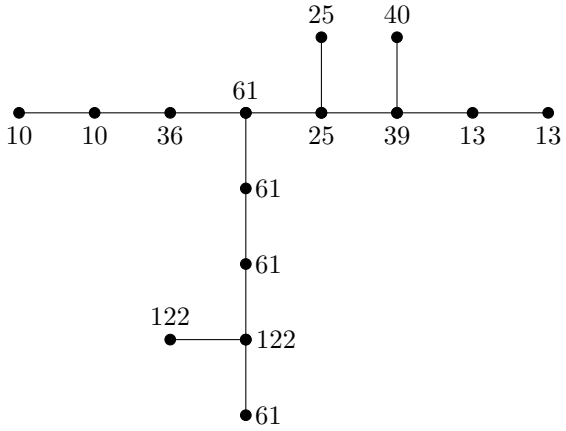
$$v(a_2) = [11\ 19\ 13\ 13\ 25\ 36\ 61\ 61\ 61\ 61\ 122\ 122\ 25\ 39\ 40]^T,$$

$$v(a_3) = [10\ 10\ 19\ 27\ 30\ 40\ 70\ 70\ 70\ 70\ 140\ 140\ 31\ 49\ 49]^T.$$

Therefore, $v_p = \min_i\{v_p(f_i)\}$ for $p \in \overline{K}$ is:

$$v = [10\ 10\ 13\ 13\ 25\ 36\ 61\ 61\ 61\ 61\ 122\ 122\ 25\ 39\ 40]^T.$$

The corresponding weighted cluster $\overline{\mathcal{K}} = (\overline{K}, v)$ is represented using the dual graph:



iii) There are two missing free base points. The cluster K' is given by the proximity matrix:

$$P_{K'} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix}.$$

The virtual values are:

$$v(a_1) = [10\ 10\ 14\ 14\ 28\ 40\ 70\ 72\ 74\ 75\ 150\ 151\ 28\ 42\ 42\ 42\ 42]^T,$$

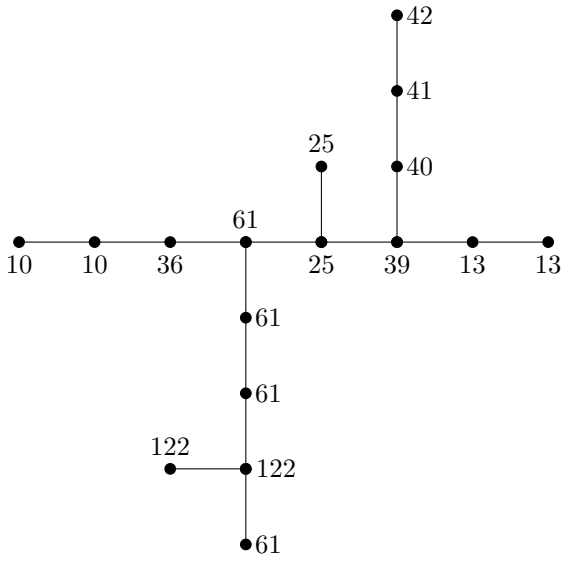
$$v(a_2) = [11\ 19\ 13\ 13\ 25\ 36\ 61\ 61\ 61\ 61\ 122\ 122\ 25\ 39\ 40\ 41\ 42]^T,$$

$$v(a_3) = [10\ 10\ 19\ 27\ 30\ 40\ 70\ 70\ 70\ 70\ 140\ 140\ 31\ 49\ 49\ 49\ 49]^T.$$

Thus, we have

$$v = [10\ 10\ 13\ 13\ 25\ 36\ 61\ 61\ 61\ 61\ 122\ 122\ 25\ 39\ 40\ 41\ 42]^T,$$

and the corresponding weighted cluster $\mathcal{K}' = (K', v)$ is represented by



iv) There are four missing satellite base points. The cluster K'' is given by:

$$P_{K''} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix}.$$

The virtual values are:

$$v(a_1) = [10\ 10\ 14\ 14\ 28\ 40\ 70\ 72\ 74\ 75\ 150\ 151\ 28\ 42\ 42\ 42\ 42\ 50\ 60\ 70\ 80]^T,$$

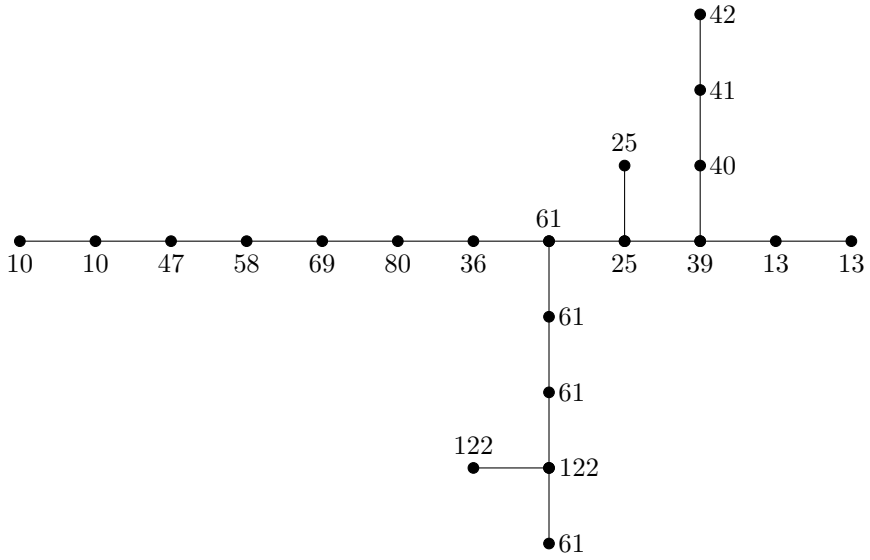
$$v(a_2) = [11\ 19\ 13\ 13\ 25\ 36\ 61\ 61\ 61\ 61\ 122\ 122\ 25\ 39\ 40\ 41\ 42\ 47\ 58\ 69\ 80]^T,$$

$$v(a_3) = [10\ 10\ 19\ 27\ 30\ 40\ 70\ 70\ 70\ 70\ 140\ 140\ 31\ 49\ 49\ 49\ 49\ 50\ 60\ 70\ 80]^T.$$

Thus, we have:

$$v = [10\ 10\ 13\ 13\ 25\ 36\ 61\ 61\ 61\ 61\ 122\ 122\ 25\ 39\ 40\ 41\ 42\ 47\ 58\ 69\ 80]^T,$$

and the corresponding weighted cluster $\mathcal{K}'' = (K'', v)$ is represented by



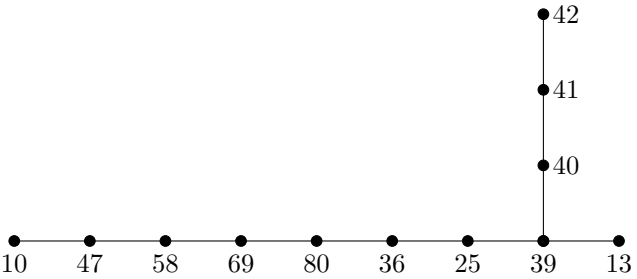
v) Using the base change formula $e^T = P_{K''} v^T$ we get

$$e = [10\ 0\ 3\ 0\ 2\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^T.$$

Thus, erasing the points with multiplicity zero, we finally obtain the weighted cluster $BP(\mathfrak{a}) = \mathcal{K} = (K, v)$ represented by the proximity matrix and the vector of values

$$P_K = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \qquad v = \begin{bmatrix} 10 \\ 13 \\ 25 \\ 36 \\ 39 \\ 40 \\ 41 \\ 42 \\ 47 \\ 58 \\ 69 \\ 80 \end{bmatrix}$$

or equivalently, by the dual graph:



5. Newton-Puiseux revisited

If we take a closer look at Algorithm 4.6 we see that all the steps can be effectively computed once we have a precise description of the weighted cluster from step ii). The aim of this section is to provide an algorithm that solves the following problem:

Given a set of elements $f_1, \dots, f_r \in \mathbb{C}\{x, y\}$, provide a method to compute the weighted cluster $\overline{K}$ associated to the reduced germ of $\xi : f = f_1 \cdots f_r = 0$ and the systems of virtual values $\{v_p(f_i)\}_{p \in \overline{K}}$, for $i = 1, \dots, r$.

We point out that in the case that f is already reduced, we can compute the cluster $\overline{K}$ using the Newton-Puiseux algorithm and Enriques' theorem [7, §1 and §5.5]. Computer algebra systems such as `Singular` [10] or `Magma` [6] can do the job. However, we are in a more general situation that requires some extra work. The Puiseux factorization theorem [7, §1.5] states that any $g \in \mathbb{C}\{x, y\}$ can be decomposed as

$$g(x, y) = u x^{\alpha_0} g_1^{\alpha_1} \cdots g_\ell^{\alpha_\ell} = u x^{\alpha_0} \prod_{i=1}^{\ell} \prod_{j=1}^{\nu_i} (y - \sigma_i^j(s_i))^{\alpha_i}, \quad \alpha_1, \dots, \alpha_l \in \mathbb{N} \tag{5.1}$$

where $u \in \mathbb{C}\{x,y\}$ is a unit, $g_1,\dots,g_\ell \in \mathbb{C}\{x,y\}$ are irreducible, $s_i \in \mathbb{C}\langle\langle x\rangle\rangle$ are Puiseux series such that $g_i(x,s_i(x)) = 0$, $\nu_i = \mathrm{ord}_y(g_i(0,y))$, and σ_i^j is the automorphism of $\mathbb{C}((x^{1/\nu_i}))$ generated by $x^{1/\nu_i} \mapsto e^{2\pi\sqrt{-1}j/\nu_i}x^{1/\nu_i}$.

From the above factorization one can compute the required cluster of singular points and systems of virtual values. It is a classical result, see [7, §5.5], that the Puiseux series $s_i, i = 1,\dots,\ell$ completely determine the cluster of singular points of η_{red}, where $\eta : g = 0$. In order to compute the virtual values $v_p(g)$ for any singular point p of η_{red} we can use the fact that v_p are valuations, thus

$$v_p(g) = \alpha_0 v_p(x) + \alpha_1 v_p(g_1) + \cdots + \alpha_\ell v_p(g_\ell). \tag{5.2}$$

In addition, the values $v_p(x), v_p(g_i), i = 1,\dots,\ell$, can also be deduced from their associated Puiseux series s_i and the cluster of singular points of η_{red}. Notice that the algebraic multiplicities α_i play their role in Equation 5.2.

The Newton-Puiseux algorithm, that traditionally has been used to obtain Puiseux decompositions, only works for reduced elements. This means that you cannot recover the algebraic multiplicities of the Puiseux series in Equation 5.1. Another problem that arises when applying the Newton-Puiseux algorithm to a product $f = f_1 \cdots f_r$ is that you cannot find which factor f_i contains each resulting Puiseux series.

To overcome such inconvenients, we will present a modified version of the Newton-Puiseux algorithm that, given a set of elements $f_1,\dots,f_r \in \mathbb{C}\{x,y\}$ not necessarily reduced or irreducible, will compute the Puiseux decomposition of the product $f = f_1 \cdots f_r$, that is, the Puiseux series of f together with their algebraic multiplicities in each of the factors $f_1,\dots,f_r$.

The Newton-Puiseux algorithm is obviously restricted to compute a partial sum of each Puiseux series in the decomposition 5.1 as the series are potentially infinite. Thus, the algorithm computes enough terms of each series so they do not share terms from a certain degree onward. In this situation we will say that the series have been pairwise *separated*. In particular, this means that a partial sum of Puiseux series s might be enough to separate s inside a factor, but not inside the whole product $f = f_1 \cdots f_r$. Hence, applying the Newton-Puiseux algorithm to the factors $f_1,\dots,f_r$ does not provide as much information as applying the Newton-Puiseux algorithm to the product. Similarly, if one obtains just the Puiseux series of the product it is not possible to recover the Puiseux decomposition of each factor. The modification of the Newton-Puiseux algorithm that we will present provides all the information needed to recover both the decomposition of each factors and the decomposition of the whole product at the same time. One of the key ingredients is the *square-free factorization*.

Definition 5.1. Let R be a unique factorization domain. The *square-free factorization* of an element $h \in R[[x]]$ is

$$h = h_1 h_2^2 \cdots h_n^n, \tag{5.3}$$

such that $h_i \in R[[x]], i = 1, \ldots, n$ are reduced, pair-wise coprime elements, and h_n is a non-unit.

Notice that some of the $h_i, i = 1, \ldots, n-1$ can be units. The non-unit factors in Equation 5.3 will be called *square-free factors* and are unique up to multiplication by a unit.

We will not explain all the details for the traditional Newton-Puiseux algorithm, for that we refer the reader to [7, 1.5]. We will just recall that it is an iterative algorithm that at the i–th step computes the i–th term of one of the Puiseux series s. The first term of $s(x) =: s^{(0)}(x_0)$ is computed from $f(x,y) =: f^{(0)}(x_0, y_0)$, and the i–th term of s is computed as the first term of $s^{(i)}(x_i) \in \mathbb{C}\langle\langle x_i \rangle\rangle$ from $f^{(i)}(x_i, y_i) \in \mathbb{C}\{x_i, y_i\}$ which are defined recursively from $s^{(i-1)}$ and $f^{(i-1)}(x_{i-1}, y_{i-1})$ by means of a change of variables.

The basic idea behind our new algorithm is to apply the traditional Newton-Puiseux algorithm to the reduced part of f, $\bar{f}$, while the square-free factors of each $f_i, i = 1, \ldots, r$ are transformed using the changes of variables given by $\bar{f}$. The Newton-Puiseux algorithm applied on $\bar{f}$ will tell when all the branches have been separated, i.e. the stopping condition. The square-free factors will encode, at the end, the algebraic multiplicities of the resulting Puiseux series in each factor.

The modified Newton-Puiseux algorithm works as follows:

- Compute the element $f = f_1 \cdots f_r$ and $\bar{f} = f/\gcd(f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y})$. Define $x_0 := x, y_0 := y, f^{(0)} := \bar{f}$, and
$$S^{(0)} := \{h_{j,k} \in \mathbb{C}\{x_0, y_0\} \mid h_{j,k} \text{ square-free factor of } f_k, k = 1, \ldots, r \text{ with multiplicity } j \in \mathbb{N}\} \tag{5.4}$$
- **Step (i)**: The i–th iteration runs as in the traditional algorithm and we compute x_{i+1}, y_{i+1} and $f^{(i+1)}$. In addition, we compute $S^{(i+1)}$ from $S^{(i)}$ in the following way:
$$S^{(i+1)} = \{x_{i+1}^{-\beta_{i,j,k}} h_{j,k}^{(i)}(x_{i+1}, y_{i+1}) \in \mathbb{C}\{x_{i+1}, y_{i+1}\} \mid x_{i+1}^{\beta_{i,j,k}+1} \nmid h_{j,k}^{(i)}(x_{i+1}, y_{i+1}), h_{j,k}^{(i+1)} \text{ non-unit}, h_{j,k}^{(i)} \in S^{(i)}\}$$
- The algorithm ends at the same step the traditional Newton-Puiseux algorithm ends for the reduced part $\bar{f}$.

In order to prove the correctness of this modification we will need the following results.

Lemma 5.2 ([7, 1.6.3])**.** *For any $j > i \geq 0$, the multiplicity of $s^{(i)}$ as Puiseux series of $f^{(i)}$ equals the multiplicity of $s^{(j)}$ as Puiseux series of $f^{(j)}$.*

In the current context the following lemma follows from the definitions.

Lemma 5.3. *Two elements of $\mathbb{C}\{x, y\}$ are coprime if and only if they share no Puiseux series and no x factor.*

Proposition 5.4. *The set $S^{(i)}$ contains the square-free factors of $f_k^{(i)}$ for any $i \geq 0$ and any $k = 1, \ldots, r$.*

Proof. By induction on $i \geq 0$. By construction, $S^{(0)}$ contains the square-free factors of $f_k^{(0)} := f^k$, for $k = 1, \ldots, r$. Assume now that $S^{(i)}$ contains the square-free factors of $f_k^{(i)}$.

If two elements of $h_{n,k}^{(i+1)}, h_{m,k}^{(i+1)}$ are not coprime, they would share a Puiseux series or an x factor, by Lemma 5.3. The x factor is not possible by definition of $S^{(i+1)}$. If they share a Puiseux series $s^{(i+1)}$, $s^{(i)}$ would be a series of $h_{n,k}^{(i)}$ and $h_{m,k}^{(i)}$, contradicting the induction hypothesis. Since $h_{j,k}^{(i)}$ is reduced so is $h_{j,k}^{(i+1)}$, by Lemma 5.2. Since Equation 5.3 still holds after applying the change of variables two both sides, the result follows. □

Proposition 5.5. *Assume $s \in \mathbb{C}\langle\langle x \rangle\rangle$ has been separated from the rest of the series of $\bar{f}$ at the i–th step of the algorithm. Then, s is a Puiseux series of $f_k \in \mathbb{C}\{x, y\}$ with algebraic multiplicity $j \in \mathbb{N}$ if and only if $h_{j,k}^{(i)} \in S^{(i)}$.*

Proof. For the direct implication, assume that s is a Puiseux series of f_k with multiplicity $j \in \mathbb{N}$. Then, s is a Puiseux series of $h_{j,k}^{(0)} \in S^{(0)}$ and no other square-free factor, by Lemma 5.3. Now, by Lemma 5.2, $s^{(i)}$ is a root of $h_{j,k}^{(i)}$ and it belongs to $S^{(i)}$ because it is a non-unit. For the converse, since s has been separated, $f^{(i)}$ has no other Puiseux series other than $s^{(i)}$ and its conjugates. By Proposition 5.4, there must be a unique $h_{j,k}^{(i)}$ square-free factor of $f_k^{(i)}$ in $S^{(i)}$. Finally, by Lemma 5.2, if the algebraic multiplicity of $s^{(i)}$ is $j > 0$ in $f_k^{(i)}$, so is the algebraic multiplicity of s in f_k. □

It follows from Proposition 5.5 that, when the algorithm stops at the i–th step after s has been separated, the set $S^{(i)}$ contains the information about the factors and the algebraic multiplicities of the Puiseux series s.

5.1. **Implementation details.** The algorithms discussed in this paper have been implemented in the computer algebra systems `Macaulay2` [11] and `Magma` [6] and they are available at

https://github.com/gblanco92/.

The `Macaulay2` implementation uses floating point arithmetic to compute the Puiseux series. This could potentially give inaccurate computations and wrong results so we implemented the same algorithms in `Magma` using algebraic field extensions. So far, all the examples the authors have tested give the same result in both implementations.

As usual when developing algorithms in computer algebra, we have to work with polynomials in $\mathbb{C}[x, y]$ instead of series in $\mathbb{C}\{x, y\}$. If we take a close look at all the steps of Algorithm 4.6 and the new formulation of the Newton-Puiseux algorithm we see that this is not an issue. Indeed, it is not a problem for the traditional Newton-Puiseux algorithm. Also, the square-free decomposition of elements of $\mathbb{C}[x, y]$ is an standard tool in computer algebra and can be computed efficiently, see for instance [12]. We also point out that, given a reduced polynomial $f \in \mathbb{C}[x, y]$, it remains reduced when viewed in $\mathbb{C}\{x, y\}$

(see [9]). Moreover, a greatest common divisor in the polynomial ring is still a greatest common divisor in the convergent series ring.

The fact that the new Newton-Puiseux algorithm works with the square-free factors of the generators, which are reduced and generally of smaller degree than the original polynomials, means that the computation can be kept efficient. Working with non-reduced elements would increase significantly the computational time of the Newton-Puiseux algorithm.

Finally, we would like to mention that Algorithm 4.6 is one of the key ingredients of the method we develop in [3] to compute the integral closure of any ideal $\mathfrak{a} \subseteq \mathbb{C}\{x, y\}$. We hope that these algorithms can be useful to people interested in the computational aspects of singularity theory. For example, our methods are very helpful in the effective computation of *multiplier ideals* (see [2], [3] and [4]).

References

[1] M. Alberich-Carramiñana, *An algorithm for computing the singularity of the generic germ of a pencil of plane curves*, Comm. Algebra **32** (2004), 1637–1646. 10, 11

[2] M. Alberich-Carramiñana, J.Àlvarez Montaner and F. Dachs-Cadefau, *Multiplier ideals in two-dimensional local rings with rational singularities*, Mich. Math. J., **65** (2016), 287–320. 21

[3] M. Alberich-Carramiñana, J.Àlvarez Montaner and G. Blanco, *Monomial generators of complete planar ideals*, preprint available at arXiv:1701.03503. 21

[4] G. Blanco and F. Dachs-Cadefau, *Computing multiplier ideals in smooth surfaces*, extended abstract for the conference "Positivity and Valuations". To appear in "Research Perspectives CRM Barcelona". 21

[5] K. Brauer, *Zur Geometrie der Functionen sweier Veranderlichen II*, Abh. Math. Sem. Hamburg, **6** (1928), 1–54. 13

[6] W. Bosma, J. Cannon and C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput., **24** (1997), 235–265. 2, 17, 20

[7] E. Casas-Alvero, *Singularities of plane curves*, London Math. Soc. Lecture Note Series, **276**, Cambridge University Press, Cambridge, 2000. 1, 2, 4, 6, 7, 8, 9, 10, 11, 17, 18, 19

[8] P. Cassou-Noguès and W. Veys, *Newton trees for ideals in two variables and applications*, Proc. Lond. Math. Soc., **108** (2014), 869–910. 3

[9] C. Chevalley, *Intersections of algebraic and algebroid varieties*, Trans. Amer. Math. Soc. **57**, (1945) 1–85. 21

[10] W. Decker, G.-M. Greuel, G. Pfister and H. Schönemann, Singular – A computer algebra system for polynomial computations. http://www.singular.uni-kl.de. 2, 17

[11] D. Grayson and M. Stillman, *Macaulay 2.* Available at: http://www.math.uiuc.edu/Macaulay2. 20

[12] D. Y. Y. Yun, *On square-free decomposition algorithms,* SYMSAC'76 Proceedings of the third ACM Symposium on Symbolic and Algebraic Computation, Yorktown Heights NY, ed. R. .D. Jenks, ACM Press, 26–35. 20

[13] O. Zariski, *On the topology of algebroid singularities,* Amer. J. Math., **54** (1932), 433–465. 13

Departament de Matemàtiques, Univ. Politècnica de Catalunya, Av. Diagonal 647, Barcelona 08028, Spain

E-mail address: Maria.Alberich@upc.edu, Josep.Alvarez@upc.edu, Guillem.Blanco@upc.edu