UNIVERSITÄT KAISERSLAUTERN
Zentrum für Computeralgebra

REPORTS ON COMPUTER ALGEBRA
NO. 33

Practical Aspects of Algorithmic Resolution of Singularities

by

A. Frühbis-Krüger, G. Pfister

October, 2004

The Zentrum für Computeralgebra (Centre for Computer Algebra) at the University of Kaiserslautern was founded in June 1993 by the Ministerium für Wissenschaft und Weiterbildung in Rheinland-Pfalz (Ministry of Science and Education of the state of Rheinland-Pfalz). The centre is a scientific institution of the departments of **Mathematics, Computer Science, and Electrical Engineering** at the University of Kaiserslautern.

The goals of the centre are to advance and to support the use of Computer Algebra in industry, research, and teaching. More concrete goals of the centre include

- the development, integration, and use of software for Computer Algebra
- the development of curricula in Computer Algebra under special consideration of interdisciplinary aspects
- the realisation of seminars about Computer Algebra
- the cooperation with other centres and institutions which have similar goals

The present coordinator of the Reports on Computer Algebra is:
Christoph Lossen (email: `lossen@mathematik.uni-kl.de`)

Zentrum für Computeralgebra
c/o Prof. Dr. G.-M. Greuel, FB Mathematik
Erwin-Schrödinger-Strasse
D-67663 Kaiserslautern; Germany
Phone: 49 - 631/205-2850   Fax: 49 - 631/205-5052
email: `greuel@mathematik.uni-kl.de`

Practical Aspects of Algorithmic Resolution of Singularities

Anne Frühbis-Krüger and Gerhard Pfister
FB Mathematik
Universität Kaiserslautern,
67653 Kaiserslautern, Germany

September 29, 2004

Abstract

Based on the development of algorithmic proofs for resolution of singularities in characteristic zero in the last two decades, it is nowadays feasible to implement these algorithms. But it is still a large step from an algorithmic proof to an algorithm which is appropriate for implementation and practical use. The aim of this article is to describe how to develop a variant of Villamayor's algorithmic proof which is suitable for practicle use. Moreover, performance of our variant allows the use of desingularisation as a tool in more comlex computational tasks. Therefore we also outline some rather direct applications of the algorithmic resolution of singularities such as the intersection matrix of the exceptional divisors, spectral numbers of a hypersurface singularity or the Denef-Loeser zeta function.

1 Introduction

The problem of resolution of singularities has stimulated mathematical research for more than a century now, ranging from first results for curves in the late 19th century (e.g. [4]) through the case of surfaces ([17], [19], [20], etc.) to the famous proof of Hironaka for the general case in characteristic zero in 1964 [10]. But this result did not end the interest in this problem. In particular, two branches of research developed over the last decades: first of all, the case of positive characteristic is still an open problem and an active field of research (e.g. [11], [1], [9]); on the other hand, the fact that Hironaka's proof was highly non-constructive led to the question of the existence of algorithmic approaches in characteristic zero and to the development thereof ([2], [3],[16], [6], etc.). This in turn now raises hope that these algorithms do not only allow implementation (as described e.g. in [5]), but may be improved to the point where resolution of singularities becomes a powerful tool for practical computations in a computer algebra system.

To contribute to this last problem is the main goal of this article. To this end, we first recall well-known facts about algorithmic resolution of singularities in section 2, emphasizing those details which are crucial for understanding the practical problems which will be dealt with in the subsequent sections. In particular, section 3 is devoted to the development of a variant of Villamayor's resolution algorithm which is specially designed for computing an embedded resolution of singularities (as opposed to Villamayor's original algorithm which deals with the more difficult problem of strong factorizing desingularization). The key ingredients in this section are the choice of the appropriate notion of transform and a significant refinement of the governing invariant of Villamayor. Equipped with this theoretical adjustment of the algorithm to our problem, we are then ready to tackle the more practical problems arising from the need to keep time and memory consumption of the implemented algorithm as low as possible. In section 4, we thus deal with questions on how the representation of the object in affine charts affects computations; we describe the choice of the center and how to avoid redundant calculations; moreover, we state the algorithms for determining the governing invariant and the center of the subsequent blow-up explicitly and also specify how to implement the blow-up.

On the basis of the implementation of our algorithm it is then possible to compute several invariants directly, whose definition relies on the knowledge about a resolution of singularities. The aim of section 5 of the article is to describe the computation of some of these invariants. As the final result of our algorithm is encoded in a collection of affine charts which glue together, the first problem to be solved is the identification of the exceptional divisors. After this, we consider the computation of the intersection form of exceptional curves on a blown-up surface and their genus. In this context, it is necessary to pass from a given embedded resolution to a non-embedded one; the main problem then turns out to be understanding and describing the splitting of the exceptional divisors: First of all, a given divisor may be irreducible in the embedded situation, but its intersection with the surface may nevertheless consist of several components; on the other hand, all computations are done over $\mathbb{Q}$ and a $\mathbb{Q}$-irreducible components can be reducible over $\mathbb{C}$. To tackle this last problem, we use results of Gao and Ruppert [8]. The next application which we consider is the computation of the negative part of the spectrum of an isolated hypersurface singularity based on a description by Varchenko [15]. Finally, we explain how to compute the local and the global Denef-Loeser zeta function for a hypersurface singularity. For these last two applications alternative approaches are known: for computing the spectrum a Hodge-theoretic approach by M. Schulze [12] is implemented in SINGULAR and for the local zeta function of a function with a non-degenerate Newton boundary there is an algorithm by by Denef and Hoornaert which is implemented in MAPLE [7].

All algorithms presented in this article are implemented in SINGULAR [13] and will be available as part of the upcoming version 2.2. In appendix A, we explain some details about the implementation itself and list timings for all of the algorithms. Appendix B focuses on the special problems arising from the aim to compute a local resolution of singularities; the last to appendices contain the examples which were mentioned in appendix A in a very explicit way and explain the specialized data structures which had to be created in order to implement the algorithms.

We would like to thank D. van Straten for pointing out the importance of not only developing a program for resolution of singularities, but also of applications using it to allow users to benefit from the program. We are especially grateful to A. Melle and I. Luengo for turning our attention to the ζ-function which was very useful for testing our implementation and eventually led to several important improvements. Last, but not least, we would like to thank the computer algebra and algebraic geometry group at the University of Kaiserslautern for many fruitful discussions.

The authors are supported in part by the DFG-Schwerpunkt "Global Methoden in der komplexen Geometrie".

2 Algorithmic Resolution of Singularities

In this section, we shall fix notation and recall well-known results and constructions about resolution of singularities and, in particular, algorithmic resolution of singularities.

As the problem of desingularitzation can be rephrased as the problem of resolving singularities of a given variety or scheme X by means of an appropriate finite sequence of blow-ups, the first notions we would like to recall are clearly the different notions of transforms under a blow-up:

Definition 2.1 *Let $I \subset \mathcal{O}_W$ be a sheaf of ideals on a smooth algebraic variety W and let $\phi : \tilde{W} \longrightarrow W$ be a blow-up map*[1] *at a smooth center C with exceptional divisor H. Then*

$$\phi^*(I) = I\mathcal{O}_{\tilde{W}}$$

and

$$I_{\text{strict}} = (\phi^*(I) :_{\mathcal{O}_W} \mathcal{I}(H)^\infty)$$

are called the total transform respectively the strict transform of I. The weak transform of I is defined as I_{weak} such that the total transform may be factorized as

$$\phi^*(I) = \mathcal{I}(H)^k I_{\text{weak}}$$

(for a suitable k) and H is not a component of $V(I_{\text{weak}})$.

Geometrically speaking, the strict transform is obtained from the total transform by dropping all components which lie inside the exceptional divisor H, whereas the weak transform originates from dropping only the component which coincides with the exceptional divisor.

[1] A short explanation on how to implement a blow-up map will be given in section 4.4.

Example 2.2 *To illustrate the difference between the weak and the strict transform under a blow-up, we will now consider the blow-up of the affine variety defined by the ideal* $I = (xy, x^3 + y^3 + z^3) \subset \mathbb{C}[x,y,z] = \mathcal{O}_W$ *at the origin, that is at the center* C *defined by* $I_C = (x,y,z)$.
Then $\tilde{W} \subset \mathbb{A}^3_{\mathbb{C}} \times \mathbb{P}^2_{\mathbb{C}}$ *is the set* $\{(x,y,z;u:v:w) \in \mathbb{A}^3_{\mathbb{C}} \times \mathbb{P}^2_{\mathbb{C}} | uy - xv = uz - xw = vz - yw = 0\}$ *which can be covered by the three affine charts corresponding to the open sets* $D(u)$, $D(v)$ *and* $D(w)$. *In each of these, the above equations imply that the affine chart looks again like an* $\mathbb{A}^3_{\mathbb{C}}$.

Chart 1: $u \neq 0$

$\mathcal{I}(H) = (x)$
$\Phi^*(I) = (x^2v, x^3 + x^3v^3 + x^3w^3) \in \mathbb{C}[x,v,w]$
$I_{\text{strict}} = (v, 1 + w^3)$
$I_{\text{weak}} = (v, x + xv^3 + xw^3) = (v, 1 + w^3) \cap (v, x)$

Chart 2: $v \neq 0$

By symmetry in x *and* y *in the generators of the original ideal, the equations here are the same as in chart 1 after exchanging* x *by* y *and* v *by* u.

Chart 3: $w \neq 0$

$\mathcal{I}(H) = (z)$
$\Phi^*(I) = (uvz^2, u^3z^3 + v^3z^3 + z^3) \in \mathbb{C}[u,v,z]$
$\overline{I}_{\text{strict}} = (uv, u^3 + v^3 + 1) = (u, 1 + v^3) \cap (v, 1 + u^3)$
$\overline{I}_{\text{weak}} = (uv, u^3z + v^3z + z) = (u, 1 + v^3) \cap (v, 1 + u^3) \cap (u, z) \cap (v, z)$

In particular, we see that the strict transform does not contain any components which are contained in the hypersurface H, *whereas in the weak transform all of those components are present except the hypersurface* H *itself.*

Another notion which is ubiquitous in the formulation of resolution type statements, is the notion of a normal crossing divisor:

Definition 2.3 *A subscheme* $Y \subset X$ *of a variety* X *is called normal crossing at a point* $y \in Y$ *if* y *is a regular point of* X *and there is a regular system of parameters* $f_1, \ldots, f_k$ *for* $y \in X$ *such that* Y *is given by the equation* $f_1 \cdot \ldots \cdot f_l = 0$ *on a Zariski neighbourhood of* y.

Equipped with these definitions of normal crossing and transforms under blow-up, we are now prepared to have a closer look at resolution theorems. There exist, in fact, several kinds of resolution type theorems, of which we would like to mention the following ones in characteristic zero[2]:

Theorem 2.4 *Let* I *be a sheaf of ideals on a smooth algebraic scheme* W. *Then there exists a principalization of* I, *that is a sequence*

$$W = W_0 \xleftarrow{\pi_1} W_1 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_r} W_r$$

[2] As already mentioned in the introduction, all fields in this article are assumed to be of characteristic zero

of blow-ups $\pi_i : W_i \longrightarrow W_{i-1}$ at smooth centers $C_{i-1} \subset W_{i-1}$ such that

- A *The morphism $W_r \longrightarrow W$ is an isomorphism over $W \setminus V(I)$.*
- B *The total transform $I\mathcal{O}_{W_r}$ of I is invertible and supported on a divisor with normal crossings, that is*

$$I\mathcal{O}_{W_r} = I(H_1)^{c_1} \cdot \ldots \cdot I(H_s)^{c_s}$$

where $\{H_1, \ldots, H_s\}$ are regular hypersurfaces with normal crossings and $c_i \geq 1$ for $1 \leq i \leq s$. If $V(I)$ does not have any codimension 1 components, the set of hypersurfaces coincides with the exceptional locus of $W_r \longrightarrow W$.

- C *The morphism $(W_r, I_r) \longrightarrow (W, I)$ is equivariant under group actions.*

Theorem 2.5 *Let X be a subscheme of a smooth algebraic scheme W. Then there exists a sequence*

$$W = W_0 \xleftarrow{\pi_1} W_1 \xleftarrow{\pi_2} \ldots \xleftarrow{\pi_r} W_r$$

of blow-ups $\pi_i : W_i \longrightarrow W_{i-1}$ at smooth centers $C_{i-1} \subset W_{i-1}$ such that

- a *The exceptional divisor of the induced morphism $W_i \longrightarrow W$ has only normal crossings and C_i has normal crossings with it.*
- b *Let $X_i \subset W_i$ be the strict transform of X. All centers C_i are disjoint from $Reg(X) \subset X_i$, the set of points where X is smooth.*[3]
- c *X_r is smooth and has normal crossings with the exceptional divisor of the morphism $W_r \longrightarrow W$.*
- d *The morphism $(W_r, X_r) \longrightarrow (W, X)$ is equivariant under group actions.*

Theorem 2.6 *Let X be an algebraic variety. There exists a desingularization of X, that is there exists a variety $\tilde{X}$ and a proper birational morphism $\pi : \tilde{X} \longrightarrow X$ such that*

1. *$\tilde{X}$ is non-singular.*
2. *π is an isomorphism over the regular points of X.*

Theorem 2.7 *The statement of theorem 2.5 can be sharpened in such a way that additionally the following condition is fulfilled:*

- e *The ideal of the total transform factorizes as*

$$\mathcal{I}(X)\mathcal{O}_{W_r} = \mathcal{I}(X_r) \cdot \mathcal{L}$$

where $\mathcal{L}$ is the sheaf of ideals of a normal crossing divisor which is a natural combination of the irreducible components of the exceptional divisor of $W_r \longrightarrow W$.

[3] This is not a typographical error, it is really $Reg(X)$, not $Reg(X_i)$. This condition simply ensures that the sequence of blow-ups is an isomorphism on $Reg(X)$.

Remark 2.8 *The previous four theorems are not only of the same basic structure, they are also very closely related. Comparison of the theorems 2.4 – 2.7 yields the following implications:*

$$\begin{array}{c} \textit{strong factorizing desingularization (2.7)} \\ \Downarrow \\ \textit{principalization (2.4)} \\ \Downarrow \\ \textit{embedded desingularization (2.5)} \\ \Downarrow \\ \textit{desingularization (2.6)} \end{array}$$

Proof:

step 1: 2.7 $\Longrightarrow$ 2.4

Let I be a sheaf of ideals on a smooth algebraic variety W. Let X be the subscheme of W defined by I. By 2.7 there is a sequence of blow-ups at smooth centers such that the morphism $W_r \longrightarrow W$ is an isomorphism over $Reg(X)$ by property (b) and $(W_r, I_r) \longrightarrow (W, I)$ is equivariant by property (d). Moreover, the total transform factorizes according to property (e) as

$$\mathcal{I}(X)\mathcal{O}_{W_r} = \mathcal{I}(X_r) \cdot \mathcal{L}.$$

If all components of X_r are hypersurfaces, then properties (a) and (c) ensure that this product has the desired structure. If there are components of X_r of lower dimension, then (a) and (c) ensure that each of these lower-dimensional components which do not intersect by the smoothness of X_r is a center satisfying the condition (a). Hence, we can obtain a principalization by blowing-up in these additional centers; as these blow-ups are again isomorphisms over $W \setminus V(I)$, the required properties (A) – (C) hold.

step 2: 2.4 $\Longrightarrow$ 2.5

Let X be an irreducible subscheme of a smooth scheme W. Then 2.4 ensures the existence of a finite sequence of blow-ups leading to a principalization of $\mathcal{I}(X)$ at smooth centers $C_i \subset W_i$. Then in the course of the principalization, there exists a step at which the strict transform X_i of X is used as the center of the following blow-up. At this point, X_i is already smooth and has normal crossings with the exceptional divisors (which is (c)); hence we may stop here. All previous centers have been inside of the corresponding X_j and since blowing-up at a smooth point of the original variety X (that is the point identified with it in X_j) would not improve the situation anyway, we may avoid blow-ups at this type of centers.
For reducible X, basically the same argument is the main part, but we have to be a little bit more careful: Again we follow the steps of the principalization of $\mathcal{I}(X)$ up to the step where the center is a whole component of X (which is not an embedded component); this component is now smooth and normal crossing with the exceptional divisors. We may

harmlessly skip this blow-up and proceed with the subsequent steps of the principalization. The same considerations apply to the other components as soon as it is their turn to be used as a center. Eventually, we only need to make sure that the components of the strict transform are separated by further blow-ups if necessary.

step 3: 2.5 $\Longrightarrow$ 2.6
Every algebraic variety can locally be embedded into affine space. The only critical point for this step is to keep track of the centers in the different charts to ensure the glueing of the charts works after each blow-up step and, in particular, at the end of the process.

$\square$

Remark 2.9 *For obtaining the product structure stated in the theorems 2.4 and 2.7, the weak transform of the ideal I is obviously the more natural notion of a transform to be considered, as the weak transform originates from the decomposition of the total transform into a product.*
For the more geometric statements 2.5 and 2.6, however, the natural choice is the strict transform, collecting all those components which do not lie inside the exceptional locus.
This observation may look rather trivial, but it will become vital for understanding the differences between Villamayor's algorithm [16] and the variant presented here in 3.

The statements about resolution of singularities, which we considered up to this point, do not give any hint on how to find the appropriate centers in an algorithmic way. The usual approach to this task is to assign to each point of the given subscheme an appropriate invariant and use the locus of the maximal value of the invariant as the center for the next blow-up.

To state this in a more formal way, some technical definitions are necessary. In particular, we need to be able to describe the current situation at an arbitrary fixed moment of the resolution process in an appropriate way and we need the tools to algorithmically describe the choice of the center:

Definition 2.10 *Let b be a positive integer. Let W be a smooth pure-dimensional algebraic variety of some dimension d, $X \subset W$ a reduced subscheme*[4]*. Let $E = \{E_1, \dots, E_k\}$ be a set of smooth normal crossing hypersurfaces in W, at each point $x \in X$ equipped with a decomposition of E into d subsets*[5]*. In this situation, we will call the tuple $\mathcal{B} = (W, X, b, E)$ a basic object.*

$Sing_b(X)$, the b–singular locus of a basic object (W, X, b, E) is defined as the closed set of all points at which the order of the ideal of X is at least b, i.e.

[4]Throughout this article, we assume additionally that X is equidimensional. Without this condition, further preparation steps would be needed before entering the resolution algorithm.

[5]For two different points $x_1, x_2 \in X$ these decompositions do not necessarily coincide

$\max\{m \in \mathbb{N} \mid \mathcal{I}_{(X,x)} \subset \mathfrak{m}^m_{(W,x)}\}$.

A blow-up of a basic object $\pi : (W', X', b, E') \longrightarrow (W, X, b, E)$ *is a blow-up* $W' \longrightarrow W$ *such that* X' *is the strict (resp. weak – depending on the context) transform of* X *and* E' *consists of the strict transforms of the elements of* E *and additionally a new element, the exceptional locus of the blow-up.*

Two basic objects (W, X, b, E) *and* (W', X', b', E') *are called isomorphic, if there is an isomorphism from* W *to* W' *inducing an isomorphism of their* $b-$ *resp.* $b'-$*singular loci and isomorphisms from the elements of* E *to those of* E' *respecting the decompositions into subsets.*

Actually, the notation (W, X, b, E) for a basic object hides the decomposition of E into subsets which is a vital ingredient to desingularization algorithms. For considering specific algorithms, we will therefore often regard E as an ordered set, ordered by the 'moment of birth' of the hypersurfaces E_i, and write (W, X, b, E, v) for a basic object where $v \in \mathbb{Z}^{d-1}$ satisfies $0 \leq v_1 \leq \ldots \leq v_{d-1} \leq k$. Using this v, we obtain a decomposition of E into d subsets $E^{(i)} = \{E_j | v_{i-1} \leq j < v_i\}$, where, of course, $v_0 := 0$.

On the other hand, the purpose of the integer b in the definition of a basic object did not really become clear up to this point. Nevertheless, it will play the key role in deciding whether a basic object is already resolved, as will be obvious as soon as we state what an algorithmic resolution should be (see 2.12 below, in particular condition (a)).

Definition 2.11 *Let* $(\mathfrak{I}, \leq)$ *be a totally ordered set.*
A family of functions[6]

$$f_{(W,X,b,E)} : X \longrightarrow \mathfrak{I},$$

which is equivariant under isomorphism of basic objects, is said to be governing a blow-up $\pi : (W_1, X_1, b, E_1) \longrightarrow (W_0, X_0, b, E_0)$*, if the following conditions hold:*

(a) The set of points $\mathrm{Max} f_{(W_0,X_0,b,E_0)} \subset X_0$*, where* f *takes its maximal value* $\max f_{(W_0,X_0,b,E_0)}$*, is a closed subset of* W_0*.*

(b) $\mathrm{Max} f_{(W_0,X_0,b,E_0)}$ *is a permissible center, that is, it is regular, has normal crossings with* E_0 *and is disjoint from the set of points* $\{x \in X_0 | x \notin Sing_b(X_0), x \notin E_{0_i} \ \forall 1 \leq i \leq \#E_0\}$*.*

(c) $\mathrm{Max} f_{(W_0,X_0,b,E_0)}$ *is the center of the blow-up* π*.*

(d) $\max f_{(W_0,X_0,b,E_0)} > \max f_{(W_1,X_1,b,E_1)}$*.*

[6] By the notation $f_{(W,x,b,E)}$ we want to emphasize that the function itself depends on the whole basic object, not just on X, although it assigns values to the points of X. The reason for this slightly strange notation is that later on it will be necessary to talk about the locus inside of X where the function takes its maximal value.

(e) *(compatibility with open restrictions)* $f_{(W_0,X_0,b,E_0)}(x) = f_{(W_1,X_1,b,E_1)}(x)$ *for each point* $x \in X_0 \setminus \mathrm{Max} f_{(W_0,X_0,b,E_0)}$*, where* $x \in X_1$ *is identified with the corresponding point in* X_0 *by means of the fact that the blow-up is an isomorphism outside of the center.*

Equipped with these technical definitions, we are now in the position to put the idea of algorithmic resolution of singularities by always blowing up at the worst points into the following more formal definition:

Definition 2.12 *Let* $(\mathfrak{I}, \leq)$ *be a totally ordered set. An algorithm of resolution of basic objects consists of a family of functions*

$$f_{(W,X,b,E)} : X \longrightarrow \mathfrak{I}$$

such that for any given basic object (W_0, X_0, b, E_0) *there is a sequence of blow-ups of basic objects*

$$\ldots \longrightarrow W_n \longrightarrow \ldots \longrightarrow W_1 \longrightarrow W_0$$

governed by the family of functions $f_{(W,X,b,E)}$*, fulfilling the following conditions*

(a) *There is an index* N *depending on the basic object such that* $Sing(X_n, b) = \emptyset$.

(b) *If* X_0 *is a regular pure-dimensional subscheme of dimension* r*,* $b = 1$ *and* $E_0 = \emptyset$*, then there is a value* $\mathfrak{s}(r) \in \mathfrak{I}$ *such that* $f_{(W_0,X_0,b,E_0)}(x) = \mathfrak{s}(r)$ *for all* $x \in X_0$.

Remark 2.13 *These last two definitions have now set up the general framework in which an algorithm for resolution of singularities may be specified. The key point of specifying an algorithm is the definition of the corresponding family of functions* $f_{(W,X,b,E)}$*, which indicates how 'bad' each point of* X *is. The basic approach for defining such a family of functions is by induction on the dimension of* W*. To this end, the function is defined explicitly in the case* $\dim(W) = 2$ *as the base of the inductive definition; for higher dimensional basic objects, say* $\dim(W) = d$*, the first part of the function and an auxilliary basic object of dimesion* $d - 1$ *are defined explicitly and the function is then obtained by combining the first part and the function on the auxilliary basic objects in a suitable way.*

Before we can specify such a family of functions explicitly, we, hence, need to give a construction to define suitable auxilliary objects and we need to find simpler functions which can be used building blocks. Of course, the choice of the auxilliary objects and of the building blocks to be used depends on the specific algorithm. However, the three main types of functions which can serve as building blocks are the following:

Definition 2.14 *Let* (W, X, b, E) *be a basic object.*

(a) The order of an ideal $\mathcal{I} \subset \mathcal{O}_{W,x}$ *at* $x \in X$ *is defined as*

$$ord_x(\mathcal{I}) := \max\{m \in \mathbb{N} | \mathcal{I} \subset \mathfrak{m}_{W,x}^m\}$$

(b) The Hilbert-Samuel function of X *at* x *is defined as*

$$H_{X,x}(l) := length(\mathcal{O}_{X,x}/\mathfrak{m}_{X,x}^{l+1})$$

Comparison of the Hilbert-Samuel function of two different points is done lexicographically in this context, that is comparison is done at the entry where the very first difference occurs.

(c) Let $E^{(1)}, \ldots, E^{(d)}$ *be the decomposition of* E *into subsets belonging to the basic object. Then*

$$n_x(E^{(i)}) := \#\{E_j \in E^{(i)} | x \in E_j\}.$$

Remark 2.15 *It is a well-known fact (see e.g. [2], [16]) that all of these functions are upper-semicontinuous in Zariski topology and that each of them does not increase if the basic object is blown-up at a permissible center. These two properties ensure that (a) and (d) of 2.11 do not contradict the use of these functions as building blocks.*

Although we will concentrate on Villamayor's resolution algorithm in this article, this is the right moment to at least mention the basic strategy of the other well-known algorithmic approach, the one of Bierstone and Milman ([2], [3]...), and say a few words on how it does or does not fit into the framework we have been setting up:
One important difference to the approach, we have been considering up to this point, is that they do not use a single integer b, but assign to each of the (fixed) generators f_i of $\mathcal{I}_{X,x}$ and a separate integer b_i - the combination of all of these basically serve the same purpose as the integer b in the above framework.
Bierstone and Milman's governing family of functions takes its values in a lexicographically ordered set of tuples of which the first entry is an infinite sequence of integers (again with lexicographic comparison), all the following entries are rationals resp. integers. The first entry originates from the fact that the first building block used for the functions is the Hilbert-Samuel function, the remaining entries correspond to building blocks which are basically of the types n_x and ord_x of the original object resp. of auxilliary objects.
At each of the blow-up steps, they use the strict transform of all objects to be considered.

To end this summary of known results, we shall now consider the basic constructions[7] behind Villamayor's algorithm ([6], [16]...) in more detail, since the

[7]This description of Villamayor's algorithm is included to allow readers direct comparison between Villamayor's algorithm and our variants and is not meant to be complete; instead we focus on the main ideas which we will use and modify for our variants later on. In particular, we will neither explain the background of the special treatment of codimension 1 components of the locus of maximal order nor the postprocessing algorithm.

goal of the following section will be to create a variant of this algorithm which is suitable for implementation and applications.

According to 2.13, formulating the algorithm may be done by specifying two objects for any given basic object (W, X, b, E, v), namely:

- the definition of the part of the function corresponding to the dimension $\dim(W) =: d$:

$$\begin{array}{rcl} f_{(W,X,b,E,v)}: & X \longrightarrow & \mathbb{Q}^{2d-2} \\ & x \longmapsto & (\frac{ord_x(\mathcal{I}_{(X,x)})}{b}, n_x(E^{(1)}), \text{entries of aux. objects}) \end{array}$$

  where the subset $E^{(1)} := \{E_i \in E \mid 0 \leq i \leq v_1\}$.

- the construction of the first auxilliary object in a two step process:

  (a) Define the following short-hand notations:

$$\begin{array}{rcl} b' & := & \max_{x \in X} \; ord_x(\mathcal{I}_{(X,x)}) \\ m' & := & \max_{x \in X} \; \{n_x(E^{(1)}\} \\ \mathcal{I}_{X'} & := & \mathcal{I}_X + \prod_{i_1 < \ldots < i_m} \sum_{j=1}^{m} \mathcal{I}(E_{i_j}) \\ E' & := & E \setminus E^{(1)} \\ v' & := & (v_2, \ldots, v_{d-1}) \in \mathbb{Z}^{d-2} \end{array}$$

  (b) Moreover, let J be a sheaf of ideals in $\mathcal{O}_W$. We define $\Delta(J) \subset \mathcal{O}_W$ as the sheaf of ideals which is locally generated by

$$\{g_i | 1 \leq i \leq s\} \cup \{\frac{\partial g_i}{\partial x_j} | 1 \leq i \leq s, 1 \leq j \leq d\},$$

  where, at a given closed point $w \in W$, $x_1, \ldots, x_d$ is a regular system of parameters for $\mathcal{O}_{W,w}$ and $g_1, \ldots, g_s$ are a set of generators for J_w. $\Delta^i(J)$ is then inductively defined as $\Delta(\Delta^{i-1}(J))$.

  (c) Let $Z \subset W$ be a closed, smooth subscheme such that Z intersects all $E_i \in E'$ transversally (condition C1)and $E' \cap Z := \{E_i \cap Z | E_i \in E'\}$ have normal crossings (condition C2).[8] Then the so-called coefficient ideal is defined as

$$Coeff_Z(\mathcal{I}_{X'}) = \sum_{i=0}^{b'-1} (\Delta^i(\mathcal{I}_{X'})\mathcal{O}_Z)^{\frac{b'!}{b'-i}}$$

[8] Note that such a hypersurface does not always exist globally. If it does not exist globally, there is an open covering of W such that on each open set there exists such a hypersurface. This passing to the open covering then leads to Villamayor's notion of general basic objects and is not vital for the understanding of the general ideas.

and the auxilliary object can be stated as $(Z, C, c, E' \cap Z, v')$, where C is the subscheme of Z corresponding to the strict (resp. weak) 'transform' [9] $(...(Coeff_Z(\mathcal{I}_{X'}) : E^{\infty}_{v_1+1}), \ldots : E^{\infty}_{\#E})$ and $c = b'!$.

Additionally, it is necessary to state how the new basic object and the corresponding lower dimensional auxilliary objects arise from a blow-up at the maximal locus of the function, in particular how the decomposition of E is affected by the blow-up: Let $\pi : (W_1, X_1, b, E_1, v^{(1)}) \longrightarrow (W_0, X_0, b, E_0, v^{(0)})$ be the blow-up governed by the family of functions as specified above, that is X_1 is the weak transform[10] of X_0, E_1 contains $\#E_0 + 1$ elements and the i-th element of E_1 is the strict transform of the i-th element of E_0 for all $i \leq \#E_0$, whereas the last element of E_1 is the exceptional hypersurface of the blow-up π. To define the tuple $v^{(1)}$, we first observe that $v^{(1)}_{i+1}$ is related to data concerning the i-th auxilliary object; in particular,

$$v^{(1)}_1 = \begin{cases} v^{(0)}_1 & \text{if} \quad Sing_{b'}(X_1) \neq \emptyset \\ \#E_1 & \text{if} \quad Sing_{b'}(X_1) = \emptyset \end{cases}$$

Of course, $v^{(1)}_1 = \#E_1$ implies $v^{(1)}_i = \#E_1$ for all $i \leq d-1$; in this case all auxilliary objects are recomputed in the same way as described before.
If, however, $v^{(1)}_1 = v^{(0)}_1$, the first auxilliary object arises from the original first auxilliary object by the blow-up governed by the family of functions in the same way as the new object arose from the original one. This construction, in turn, leads to defining $v^{(1)}_2$ analogously to $v^{(1)}_1$ - this time using the c'-singular locus of the auxilliary object. By iteration this defines the whole tuple $v^{(1)}$ and the whole set of auxilliary objects for $(W_1, X_1, b, E_1, v^{(1)})$.

Remark 2.16 *At this point, it is important to observe that generating the auxilliary objects for an object arising from a blow-up in the same way as for the original object, we would not obtain the same set of auxilliary objects:*
Consider, for instance, the Whitney umbrella defined by the following affine basic object

$$\mathfrak{A} = (\mathbb{A}^3_{\mathbb{C}}, V(z^2 - x^2 y), 1, \emptyset, (0,0))$$

having as the first auxilliary object

$$\mathfrak{B} = (V(z), V(x^4, x^2 y), 1, \emptyset, (0))$$

and the blow-up at center $V(x,y,z)$; the value of the governing function at the center is $(2,0,3,0)$.

[9] Actually, Villamayor's algorithm only uses this construction for the case $E' = \emptyset$. We formulate it more generally here because we will need this in the following two sections. Although this is not really a transform under a blow-up, it is constructed by iterated ideal quotients where the iteration stops as soon as the condition for a weak resp. a strict transform is satisfied. Because of this, we will also sometimes say the weak Coeff-ideal or the strict Coeff-ideal.

[10] As mentioned previously, the goal of Villamayor's algorithm is to compute a strong factorizing desingularization and hence the appropriate notion of transform in this context is the weak transform.

Of the three affine charts of the blown-up object, the interesting one is the following:

$$\mathfrak{A}_1 = (\mathbb{A}^3_{\mathbb{C}}, V(z^2 - x^2y), 1, \{V(y)\}, (0,1))$$

Here the weak transform of the first auxilliary object is

$$\mathfrak{B}_1 = (V(z), V(x^2), 1, \{V(y)\}, (1))$$

leading to the functing value $(2,0,2,1)$*, whereas the first auxilliary object of* $\mathfrak{A}_1$ *is*

$$\mathfrak{B}_1^{'} = (V(z), V(x^4, x^2y), 1, \{V(y)\}, (1))$$

implying function value $(2,0,3,1)$.

3 Variants of Villamayor's Resolution Algorithm

The algorithm of Villamayor relies on the use of the weak transform of the object X respectively the auxilliary objects in lower dimensions. Geometrically speaking, the main difference is that, passing from the total transform to the strict transform, all components lying inside the new exceptional hypersurface are dropped, whereas passing to the weak transform only involves dropping the new exceptional hypersurface itself, but not components inside of it. The computational effect of this difference is that, in general, the degrees of the generators of the ideal of the weak transform are higher than the ones of the strict transform – making the subsequent computations harder. Thus it may be regarded as a (practical) drawback, that Villamayor's algorithm uses the weak transform. Using the strict transform would also remove the peculiarity (see 2.16) that an auxilliary object of a transformed object is, in general, not obtained in the same way as an auxilliary object of an original object; more precisely, the following lemma states that for the strict transform, this difference does not occur:

Lemma 3.1 [11] *Let* $\pi : (W_1, X_1, b, E_1) \longrightarrow (W, X, b, E)$ *be a blow-up of basic objects and let* $\mathcal{I} \subset \mathcal{O}_W$*. Then*

$$(\Delta(\mathcal{I}\mathcal{O}_{W_1} : \mathcal{I}(H)^\infty) : \mathcal{I}(H)^\infty) = (\Delta(\mathcal{I})\mathcal{O}_{W_1} : \mathcal{I}(H)^\infty),$$

where $\mathcal{I}(H)$ *is the ideal sheaf of the exceptional divisor arising from the blow-up* π*.*
In particular, the strict transform of the Coeff-ideal and the ('strict') Coeff-ideal of the strict transform coincide.

Proof: Let $\delta : \mathcal{O}_W \longrightarrow \mathcal{O}_W$ be a globally defined derivation and let $\mathcal{J} \subset \mathcal{O}_W$ be an invertible sheaf of ideals. Let $p \in W$ be a closed point of the center of the blow-up π and let $\{x_1, \dots, x_n\} \subset \mathcal{O}_W$ be a regular system of parameters.

[11] This lemma is closely related to the lemma 19.1 of [6] which in turn attributes the idea to J. Giraud

W.l.o.g we may assume that the center of the π is locally defined by the ideal $(x_1,\ldots,x_s)\subset\mathcal{O}_{W,p}$. As we can restrict our considerations to an affine neighbourhood $U\subset W$ of p such that $x_1,\ldots,x_s$ are global sections of $\mathcal{O}_U$ and such that J is generated by a global section f, we may also assume for simplicity of notation that $W=U$.

step 1: $\mathcal{I}(H)\cdot\delta$ is an invertible sheaf of derivations on W_1
Now, W_1 may be covered by affine charts U_i such that we have in each of the charts:

$$\mathcal{O}_{U_i}\cong\mathcal{O}_W[y_1,\ldots,\hat{y}_i\ldots,y_s]/(x_j-x_iy_j|1\leq j\leq s,j\neq i)$$

where $\hat{y}_i$ indicates that y_i is missing from the list $y_1,\ldots,y_s$; in this chart $\mathcal{I}(H)=(x_i)$. Now we can consider $\mathcal{I}(H)\cdot\delta$ in this chart: first of all, we observe that $\delta(x_j)$ is, of course, already defined for all $1\leq j\leq n$. For $1\leq j\leq s,\ \ j\neq i$, we inherit the equality

$$x_i\cdot\delta(y_j)=\delta(x_j)-y_j\cdot\delta(x_i)$$

from the fact that $x_j-x_iy_j=0$; hence $\mathcal{I}(H)\cdot\delta|_{U_i}=x_i\cdot\delta:\mathcal{O}_{U_i}\longrightarrow\mathcal{O}_{U_i}$ and therefore $\mathcal{I}(H)\cdot\delta$ is an invertible sheaf of derivations on W_1.

From now on we focus on one chart, say U_i. In this chart, we may consider the total transform[12] J_1 of J which is generated by the global section $f_1=x_i^b\cdot g$ on U_i for a suitable integer b and a suitable g which is not divisible by x_i.

step 2: $((f_1,\delta(f_1)):x_i^\infty)=(g,\delta(g))$ on U_i
This assertion can be checked by direct computation:

$$\delta(f_1)=\delta(x_i^b)\cdot g+x_i^b\cdot\delta(g)$$

implies that

$$\begin{aligned}((f_1,\delta(f_1)):x_i^\infty)&=((x_i^b\cdot g,\delta(x_i^b)\cdot g+x_i^b\cdot\delta(g)):x_i^\infty)\\&=((g,g+x_i\cdot\delta(g)):x_i^\infty)\\&=(g,\delta(g)).\end{aligned}$$

step 3: $(\Delta(J)\mathcal{O}_{W_1}:x_i^\infty)=(\Delta(J_1):x_i^\infty)$ on U_i
Again, we consider the situation in each chart U_i and see by direct computation:

$$\begin{aligned}(\Delta(J_1):x_i^\infty)&=((f_1,\frac{\partial f_1}{\partial x_i},\frac{\partial f_1}{\partial y_1},\ldots,\frac{\partial f_1}{\partial y_s},\frac{\partial f_1}{\partial x_{s+1}},\ldots\frac{\partial f_1}{\partial x_n}):x_i^\infty)\\&=((f_1,\frac{\partial f_1}{\partial x_i},\frac{\partial f_1}{\partial x_1}\cdot\underbrace{\frac{\partial x_1}{\partial y_1}}_{=x_i},\ldots,\frac{\partial f_1}{\partial x_s}\cdot\underbrace{\frac{\partial x_s}{\partial y_s}}_{=x_i},\frac{\partial f_1}{\partial x_{s+1}},\ldots\frac{\partial f_1}{\partial x_n}):x_i^\infty)\end{aligned}$$

On the other hand,

$$(\Delta(J)\mathcal{O}_{W_1}:x_i^\infty)=\left(\left(f_1,\left(\frac{\partial f}{\partial x_1}\right)_1,\ldots,\left(\frac{\partial f}{\partial x_n}\right)_1\right):x_i^\infty\right)$$

[12]To keep notation short, a lower index $_1$ attached to an object living in W indicates the total transform throughout this proof.

and

$$\frac{\partial f_1}{\partial x_j} = x_i \cdot \left(\frac{\partial f}{\partial x_j}\right)_1$$

for all $1 \leq j \leq n$, $i \neq j$ respectively

$$\frac{\partial f_1}{\partial x_i} = x_i \cdot \left(\frac{\partial f}{\partial x_i}\right)_1 + \sum_{j \neq i} \underbrace{x_j \cdot \left(\frac{\partial f}{\partial x_j}\right)_1}_{=\frac{\partial f_1}{\partial x_j}}.$$

Combining these facts, we directly obtain the proof of the claim for the case of the invertible sheaf J.

step 4: General case

As we have done for J in step 1, we can reduce to the case that $\mathcal{I} = (g_1, \ldots, g_t) \subset \mathcal{O}_W$ and apply the steps 2 and 3 to each of the g_i, proving the general case. Now the Coeff-ideal is constructed by means of suitable powers of ideals of type $\Delta^k(\mathcal{I})$ and hence also the assertion about the Coeff-ideals is proved.

$\square$

Remark 3.2 *If the goal of the calculations is a factorizing desingularization or a principalization of an ideal, the problem itself requires the use of the weak transform instead of the strict as we have seen in 2.9. For embedded resolution of singularities, on the other hand, where we are only interested in the resulting strict transform at the end of the sequence of blow-ups (and, of course, the exceptional locus), the use of the strict transform seems to be the natural approach.*

Clearly, the first idea is to substitute the weak transform by the strict in the algorithm of Villamayor, but to leave the governing family of functions unchanged. Properties (a), (d) and (e) of a blow-up governed by the family of functions are clearly satisfied, whereas properties (b) and (c) which are actually the central part of this notion are violated as the following example illustrates:

Example 3.3 *Consider the basic object*

$$\mathfrak{A} = (\mathbb{A}^3_{\mathbb{C}}, V(z^2 - x^3y^3), 1, \{V(x), V(y)\}, (0, 2)).$$

Its first auxilliary object is obtained by passing to the hypersurface $V(z)$ and forming the iterated ideal quotients of the ideal $(x^3y^3) + (x^2y^3, x^3y^2)^2$ first with the ideal (x) and then with (y) corresponding to the exceptional divisors. Obviously, the iterated ideal quotients lead to the ideal (1), which implies that the maximal value of the governing function is $(2, 0, 0, 0)$, since clearly the order of the original ideal is 2 and there are no exceptional divisors in the first subset of E . The locus of maximal value is given by the radical of the ideal $\Delta(z^2 - x^3y^3) = (z, x^2y^3, x^3y^2)$, that is by the ideal (z, xy) which obviously has a singular point at $V(x, y, z)$.

As this example shows, we may not drop all components of the weak transform of the auxilliary object which do not occur in the strict transform. Therefore we need to refine the family of functions in the following way to ensure that it leads to premissible centers: We plug in an additional entry after each entry of type n_x. Denoting by $X_1, \ldots, X_t$ the components of the upcoming Coeff-ideal (formed using weak transforms), which contain the point at which we want to evaluate the function, this new entry may be stated as follows:

$$(-c; i_1, \ldots, i_c)$$

where

$$c := \min_{1 \leq j \leq t} \left\{ s \in \mathbb{N} \;\middle|\; \exists i_1, \ldots, i_s \text{ s.th. } \begin{array}{c} X_j \subset \bigcap_{1 \leq k \leq s} E_{i_k} \\ X_j \not\subset \bigcup_{m \notin \{i_1, \ldots, i_s\}} E_m \end{array} \right\}$$

and $i_1 < i_2 < \ldots < i_c$ are the indices of the corresponding exceptional hypersurfaces. As isomorphisms of basic objects do not affect the decomposition of the set E into subsets, this refined family of functions is again equivariant and we may proceed to check the conditions (a) to (e) of a blow-up governed by a family of functions:

Lemma 3.4 *The above family of functions satisfies the conditions (a) – (e) of 2.11.*

Proof:

ad (a) Since the additional entries only select components of the Coeff-ideal which is about to be formed, the locus where f takes its maximal value is still a closed subset of W.

ad (b) Consider the auxilliary object of lowest dimension. For this object the maximal locus of the invariant is determined in the same way as in Villamayor's algorithm. Hence we have a permissible center for this auxilliary object. In particular, it is regular.
Since already the 1–singular locus of the first auxilliary object is disjoint from the set of points which are not allowed to be altered, also the third part of the condition is clearly satisfied.
For the normal crossing property, we need to observe that the locus of maximal value of f is normal crossing with the elements of the last subset of E, because it is permissible for the auxilliary object of lowest dimension. But all the other subsets of E are being taken into account by considering the maximal locus of the entries of type n_E for each of the auxilliary objects of higher dimension, that is by intersecting with the suitable E_i in exactly the same way as in the construction of Villamayor which in turn allows us to inherit the normal crossing property here as well.

ad (c) This condition is related to the blow-up and not a condition for the family of functions. So there is nothing to do here.

ad (d) As the maximal value of f on the auxilliary object of lowest dimension is determined in the same way as in Villamayor's algorithm, it will certainly drop. Therefore, this property is proved except in one case, namely in the case that the X of the auxilliary object of lowest dimension is used as the center of the blow-up and hence completely 'disappears' due to this blow-up. In this situation, the component of the lowest-dimensional Coeff-ideal, which was used to create this auxilliary object, does not exist any longer; we move on to the next component of this Coeff-ideal which might be corresponding to the same entry of type $-c$, but a lexicographically smaller list of indices (value dropped) or to a higher c, i.e. a lower $-c$ (value dropped). The only remaining possibility which we need to consider is that there is no other component of the Coeff-ideal left, but in this situation the value of some earlier entry has already dropped by Villamayor's arguments.
Putting all this together, we see that this change in strategy has not affected the property (d).

ad (e) This property holds by construction of the family of functions.

□

Since this new family of functions is only a refinement of Villamayor's family of functions, the properties (a) and (b) of an algorithm for resolution of singularities are inherited from Villamayor's family of functions.

Obviously, we did not make much progress towards our original goal up to now. In particular, considering the refined family of functions more closely, we see that we are actually following Villamayor's algorithm with a slight change in the selction strategy for the Coeff-ideal. But we have not (yet) made the step from weak to strict transform, nor did we eliminate the fact that the auxilliary objects after the blow-up are the transforms of the original auxlliary objects and not the auxilliary objects of the transformed object. Nevertheless, this refinement of the family of functions is vital to our approach for solving these two problems: by a further refinement of the family of functions, we exclude a large number of the components of the Coeff-ideal which allows us to get as close to the strict transform as possible, and at that point we are then able to apply the lemma 3.1.

Recall that after the first refinement the family of functions is constructed according to the following pattern:

$$(b_1, n_1; (-c_2; i_1^{(2)}, \ldots, i_{c_2}^{(2)}), b_2, n_2; (-c_3; \ldots) \ldots).$$

The second refinement step now introduces some additional factors to this pattern, which might best be understood as boolean values and lead to a pattern:

$$(b_1, n_1; \varepsilon_1^{(2)}((-c_2; i_1^{(2)}, \ldots, i_{c_2}^{(2)}), b_2, \varepsilon_2^{(2)}(n_2; \varepsilon_1^{(3)}((-c_3; \ldots) \ldots)))),$$

where $\varepsilon_j^{(i)}$ are defined as follows:
Fix some arbitrary point p at which we want to evaluate the function and denote

the auxilliary object of level $(i-1)$ by (W, X, b, E, v), then

$$\varepsilon_1^{(i)}((-c_i; i_1^{(i)}, \ldots, i_{c_i}^{(i)}), b_i, \ldots)(p) = ((-\infty), 0, 0; 0, \ldots), \text{ if}$$

there is a open subset $U \subset W$, containing p, for which the maximal value on U of the function truncated after the entries for level $(i-1)$ coincides with the value at p of the truncated function, such that

- all the components of the currently constructed Coeff-ideal (leading to level (i)) are contained in the union of the elements of E of the upcoming auxilliary object of level (i)
- the locus of this maximal value satisfies the conditions for being a permissible center in U.

If one of these conditions is not satisfied, $\varepsilon_1^{(i)}$ does not change the function, that is

$$\varepsilon_1^{(i)}((-c_i; i_1^{(i)}, \ldots, i_{c_i}^{(i)}), b_i, \ldots)(p) = (-c_i; i_1^{(i)}, \ldots, i_{c_i}^{(i)}), b_i, \ldots)(p)$$

. The philosophy behind this refinement is to ignore components of the 'weak Coeff-ideal' which are not components of the 'strict Coeff-ideal' whenever it does not lead to singular centers.

Denoting now for simplicity of notation the auxilliary object of level (i) by (W, X, b, E, v), we can define

$$\varepsilon_2^{(i)}(n_i; \varepsilon_1^{(i+1)}((-c_{i+1}; \ldots), \ldots))(p) = (0; 0, 0, \ldots) \text{ if}$$

there exists a neighbourhood $U \subset W$ containing p, for which the maximal value on U of the function truncated after b_i coincides with the value at p of the truncated function, such that

- the locus of this maximal value is of dimension one
- the locus of this maximal value is a permissible center in U
- truncating the function after n_i leads to a zero-dimensional locus of maximal value

If one of these conditions does not hold, $\varepsilon_1^{(i)}$ does not change the function, that is

$$\varepsilon_2^{(i)}(n_i; \varepsilon_1^{(i+1)}((-c_{i+1}; \ldots), \ldots))(p) = (n_i; \varepsilon_1^{(i+1)}((-c_{i+1}; \ldots), \ldots)(p).$$

This refinement of the family of functions again satisfies all conditions imposed on a family governing a blow-up resp. an algorithm of resolution of singularities as can be checked directly. In particular, the only obstacle to using the strict transform instead of the weak one was that condition (b) (permissible center) for a family governing a blow-up was not satisfied; this property is now ensured in this refined family, since the ε_i only have effect if the candidate for a center is permissible.

The last difference between our algorithm and the one of Villamayor is that we always consider the auxilliary objects generated from the original object instead of the transform of the previous auxilliary object. By lemma 3.1, this does not make any difference if all c_i are zero which appear in the maximal value of the function for the object to be considered. In all other cases, the candidate for the center, which would be chosen by using strict transforms for evaluating the functions, is not permissible because it is still singular. But this fact is encoded in the remaining components of the (weak) Coeff-ideal which are – one after the other – first resolved until they are themselves non-singular and then used as the center of a blow-up; the process stops as soon as the singularities of the original candidate of the center are resolved.[13]

4 Practical Aspects of the Resolution Algorithm

In the previous section, we considered the theoretical changes to the resolution algorithms which were necessary to optimize it for the task of embedded resolution of singularities. But still the variant of the resolution algorithm presented in the previous section is of theoretical nature; for instance, it deals with the general resp. the projective case, whereas an implementation will represent the objects by means of affine charts, it assumes that the governing function is evaluated completely at all points of the given basic object and it relies on constructions of Δ and Coeff which have only been specified locally at each point. But evaluating and constructing at each of the points of our basic object is simply not feasible. The goal of this section is to formulate the changes necessary to specify a practical (that is implementable) algorithm for the embedded resolution problem in arbitrary dimension.

As already mentioned, the first important difference between the theoretical approach and the practical implementation is that the basic objects will be represented by means of a set of affine charts $\{U_i | 1 \leq i \leq s\}$ covering W. Although this is essential for being able to compute with basic objects, it raises several problems which we need to discuss briefly before proceeding:

Patching of Center For a given basic object, the center is (according to the theoretical algorithm) specified by the locus of maximal value of the governing function on $X \subset W$. In practice, of course, the maximal locus is determined separately in each of the affine charts U_i and the question whether we can proceed by blowing up at this center in each of the charts arises. If the maximal value is the same in all charts, the answer is clearly yes; but the critical case is that the maximal values in the charts do not

[13] This argument is actually an inductive one, of which we have only mentioned the step of the induction here. For the base of the induction, we first have to consider the basic objects of dimension two for which obviously no auxilliary objects are created and hence the algorithm coincides with the one of Villamayor.

coincide. This can only occur if the locus of the overall maximal value does not meet all charts and hence certain charts stay unchanged by this blow-up and await the point in the sequence of blow-ups when the overall maximal value has dropped to the one on this chart. At this point, the locus of maximal value in this chart is then used for blowing-up this chart. Hence the patching of the center is already ensured by the governing family of functions.

Redundant calculations Each blow-up increases the number of charts to be considered and thus the number of charts meeting the locus of maximal value of the governing function may increase dramatically during the resolution process, even if the locus of maximal value is a single point. This leads to redundant calculations which can be very time- and memory-consuming. On the other hand, it is not possible to simply omit such a redundant calculation, since all blow-ups in a chart have to be done in the proper order. So the only means we have for keeping the number of redundant calculations down is by avoiding subdivisions of charts whenever possible and by dropping charts which do not contain any points of the basic object which are not covered by other charts.

At this point it is important to observe that there is a philosophical difference between Villamayor's approach using general basic objects and our approach: Villamayor formulates the sequence of blow-ups on each affine chart and checks compatibility – in fact, he is forced to do so, since he transforms the auxilliary objects instead of recomputing them at each step of the sequence of blow-ups. We, on the other hand, are allowed to generate the auxilliary objects in each step for the current main object and may hence regard a blow-up as what it is, a birational morphism of the whole algebraic variety; in particular, we may consider the restrictions to the affine charts simply as charts used in the current step. Using our approach reduces the problem of passing from one set of affine charts to a new one at each blow-up to the purely technical problem of keeping track of which point appears in which chart and how do the charts intersect.

4.1 Computing $\Delta(J)$

In section 2, the definition of $\Delta(J)$ heavily relied on using generators for $J\mathcal{O}_{W,w}$ and a regular system of parameters for $\mathcal{O}_{W,w}$ at the given closed point $w \in W$. Theoretically this is fine, but in practice it is, of course, not feasible to compute at each point of W. Here, the use of a set of generators of $J\mathcal{O}_{W,w}$ does not cause any problems, since we are working on affine charts U_i and on each chart we are specifying J by a set of generators anyway. The difficulties arise from the need for a global system of local regular parameters for $\mathcal{O}_{W,w}$ on U_i, which, in general, does not exist. Instead it is necessary to pass to a suitable open covering $\{U_{ij}\}$ of U_i such that for each U_{ij} we can find a global system giving rise to a regular system of parameters at each point of U_{ij}. This, in turn, increases the number of charts which we can avoid by recombining the results on the U_{ij} to one on U_i. More precisely, $\Delta(J)$ is determined by the following algorithm:

Algorithm Delta

Input $(g_1,\dots,g_r)$ generating $\mathcal{I}_W \subset \mathbb{C}[x_1,\dots,x_n] = \mathcal{O}_{U_i}$
$(f_1,\dots,f_s)$ generating $\mathcal{I}_X \subset \mathbb{C}[x_1,\dots,x_n]$
such that $V(\mathcal{I}_W)$ is equidimensional and regular and $\mathcal{I}_W \subset \mathcal{I}_X$.

Output $\Delta(\mathcal{I}_X) \subset \mathbb{C}[x_1,\dots,x_n] = \mathcal{O}_{U_i}$

1. if $\mathcal{I}_W = (0)$
   then return$((f_1,\dots,f_s,\frac{\partial f_1}{\partial x_1},\dots,\frac{\partial f_s}{\partial x_k}))$
2. Initialization
   $C = \{f_1,\dots,f_s\}$
   $D = (1)$
   $L1 = \{\ n-\dim(W)$ square submatrices of the Jacobian matrix of $\mathcal{I}_W$ whose determinant is non-zero$\}$ [14]
3. **while** $(L1 != \emptyset)$
   - choose $M \in L1$
     $L1 = L1 \setminus \{M\}$
   - $q = \det(M)$
   - determine an $n-\dim(W)$ square matrix A such that
     $A \cdot M = q \cdot E_{n-\dim(W)}$ [15]
   - determine components of $\Delta(J)$ not lying inside $V(q)$:

     $$C_M = C \cup \left\{ q \cdot \frac{\partial f_i}{\partial x_j} - \sum_{\substack{k \text{ row of } M \\ l \text{ column of } M}} \frac{\partial g_l}{\partial x_j} A_{lk} \frac{\partial f_i}{\partial x_k} \;\middle|\; \begin{array}{c} 1 \le i \le s, \\ j \text{ not row of } M \end{array} \right\}$$

     $C_M = sat(C_M, q)$
   - Add these components to the previously found ones:
     $D = D \cap C_M$
4. return(D)

The basic idea behind this algorithm is that W is regular and hence at each point there is at least one $(n-\dim(W))$-minor of the Jacobian matrix of $\mathcal{I}_W$ which does not vanish. So $\Delta(J)$ is computed separately on each complement of a minor of the Jacobian, then we pass to all of U_i again by saturation and combine the results of the computations on the complements.

Remark 4.1 *From the practical point of view the above algorithm still needs to be improved to avoid redundant calculations. In particular, one should first check whether there is a minor of the appropriate size which is itself an element of* $\mathbb{C}$. *In this case, the complement of the minor is the whole open set* U_i *and the other minors do not give any new contributions.*

[14] For simplicity, the row and column indices used inside the submatrices will be the ones of the corresponding rows resp. columns in the Jacobian matrix.
[15] E_j denotes the $j \times j$ unit matrix. As before, we use row and column indices corresponding to those of M for simplicity

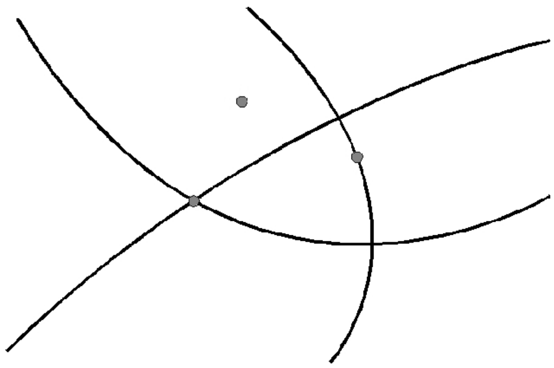

Figure 1: As an example for the problem of computing $\Delta(J)$, let us consider the situation illustrated in the above pictur: There are three minors whose determinant does not vanish (each one illustrated by one of the curves in the above picture) and $V(\Delta(J))$ consists of the three points. Then computing on the complement of just one of the minors will not provide all points of $V(\Delta(J))$, because each of the curves meets at least one point.

4.2 Computing $\mathrm{Coeff}(J, b)$

The construction of the Coeff-ideal in section 2 involved the choice of a smooth hypersurface Z subject to two conditions (C1) and (C2) on the intersections with elements of E. It was already pointed out there (in footnote 7) that such a hypersurface does not necessarily exist globally; by passing to a finite open covering, however, it is possible to find such a hypersurface for each of the open sets. Now the first idea to keep the number of open sets as low as possible is to recombine in the end, just as in the previous algorithm. Unfortunately, the auxilliary objects do depend on the chosen hypersurface, although the resulting value of the governing function at each point is independent of this choice. Therefore, we cannot just recombine directly as before; instead, we continue with the algorithm for finding the maximal locus of the governing function in each of the open sets and then (carefully) recombine those maximal loci.

Algorithm Coeff

Input basic object (W, X', b', E', v') on U_i
in the notation of section 2

Output list of pairs (U_{ij}, B_j),
where the sets U_{ij} specify an open covering of U_i and B_j is a Coeff-object for the original basic object on U_{ij}

1. iteratively determine $\Delta^1(\mathcal{I}_{X'}), \ldots, \Delta^{b'-1}(\mathcal{I}_{X'})$
2. if ((C1) and (C2) hold for one of the generators of $\Delta^{b'-1}(\mathcal{I}_{X'})$
   or for a general $\mathbb{C}$-linear combination thereof)

then • call hypersurface Z
• determine the Coeff-object $(Z, C, c, E' \cap Z, v')$
• return($\{U_i$,$(Z, C, c, E' \cap Z, v')\}$)

3. find a suitable open covering:
   choose $f_1, \ldots, f_s$ from the given set of generators of $\Delta^{b'-1}(\mathcal{I}_{X'})$ such that $\bigcap_{j=1}^{s} Sing(f_j) = \emptyset$ and such that the conditions (C1) and (C2) hold for $Z_j = V(f_j)$ on the complement U_{ij} of $Sing(f_j)$ for each j
4. for each $1 \leq j \leq s$ determine the Coeff-object $(Z_j, C_j, c, E' \cap Z_j, v'_j)$
5. return($\{(U_{ij}$,$(Z_j, C_j, c, E' \cap Z_j, v'_j)) \mid 1 \leq j \leq s\}$)

The choice of a suitable open covering in the above algorithm is the most delicate part of it. The existence of such a covering is ensured by the fact that when the order first takes the value of b' the set E' is empty and hence (C1) and (C2) are trivially fulfilled and, on the other hand, for each point x of X' there has to be at least one generator of $\Delta^{b'-1}(\mathcal{I}_{X'})$ which defines a hypersurface that is smooth at x. The construction of the centers for the following blow-ups then ensures that (C1) and (C2) also hold after these blow-ups (until the order drops of course which, in turn, puts us back into the original situation but with a lower b') as a straight forward computation shows. Therefore it is not a problem of whether such a covering exists, but simply of finding it from a given basic object without any additional reference to the previous blow-ups. The key idea here is to express 1 as a combination of the generators of the singular loci of the generators of $\Delta^{b'-1}(\mathcal{I}_{X'})$ and use the complements of those generators appearing with non-zero coefficients as the open covering – of course only if (C1) and (C2) hold, otherwise we pass to another combination expressing 1.

Remark 4.2 *In practice, the complement of a hypersurface is by far easier to handle than the complement of a lower dimensional variety, as the first only involves adding one new variable and one new equation. This advantage (in complexity) usually even outweighs the disadvantage of being forced to pass to a larger number of open sets in the covering of U_i.*

4.3 Algorithm for determining the center

In section 2, we saw that the center for the following blow-up is determined by the locus of maximal value of the governing function. In practice, it is impossible to evaluate the function at each of the infinitely many points. For determining the center, however, this is not necessary anyway. Instead, it suffices to iteratively determine the locus of maximal value of the first entry, inside of this then the one of maximal value of the second entry and so on. More precisely:

Algorithm Center

Input Basic object (W, X, b, E, v) on open set U_i

Output ideal C describing the center for following blow-up

1. determine b_1 such that $\Delta^{b_1}(\mathcal{I}_X) = (1)$, but $\Delta^{b_1-1}(\mathcal{I}_X) \neq (1)$
2. $\mathcal{M} = \{A \subset \{1, \ldots, v_1\} \mid V(\mathcal{I}_X) \cap \bigcap_{i \in A} E_i \neq \emptyset\}$
3. $C = \bigcap_{A \in \mathcal{M}} (\mathcal{I}_X + \mathcal{I}(\bigcap_{i \in A} E_i)^{b_1})$
   $v' = (v_2, \ldots, v_{d-2})$
   $E' = E \setminus \{E_1, \ldots, E_{v_1}\}$
4. if ($\sqrt{\Delta^{b_1-1}(\mathcal{I}_X)}$ permissible center and
   $\dim(\Delta^{b_1-1}(\mathcal{I}_X)) = 1$ and
   $\dim(\Delta^{b_1-1}(C)) = 0$)
   then return($\sqrt{\Delta^{b_1-1}(\mathcal{I}_X)}$)
5. L=Coeff($W, V(C), b_1, E', v'$)
   $C_0 = (1)$,$maxinv = -\infty$
6. if($\Delta^{b_1-1}(C)$ not permissible)
   then $C_1 = (1)$,$maxinv_1 = -\infty$
7. while $L \neq \emptyset$
   - choose $(U, B) \in L$
     $L = L \setminus (U, B)$
   - if(B is not empty set)
     then $D = Center(B)$
     if((value at D)> $maxinv$) [16]
     then $C_0 = D$
     $maxinv$=value at D
     if((value at D) $= maxinv$) then $C_0 = C_0 + D$
   - if($C_0 = (1)$ and $\Delta^{b_1-1}(C)$ not permissible)
     then D = component of highest function value of
     weak Coeff-ideal on U
     if((value at D) $> maxinv_1$)
     then $C_1 = D$
     $maxinv_1$=value at D
     if(value at $D = maxinv_1$) then $C_1 = C_1 + D$
8. if($C_0 = (1)$)
   then if($\Delta^{b_1-1}(C)$ permissible)
   then return(C)
   else return(C_1)
9. return(C_0)

In the above algorithm, step 1 determines $\Delta^{b_1-1}(\mathcal{I}_X)$, the locus of maximal value of the first entry of the governing function. In step 3 the candidate for the maximal locus of the first two entries, $\Delta^{b_1-1}(C)$, is being evaluated and then corrected by the refinement ε_2 in step 4 if necessary. Step 7 invokes of the

[16] We write 'value at D' for the value assigned to one and hence every point of D by the appropriate function from the governing family of functions.

recursion and also takes care of the fact that we might be forced to cover U_i by open subsets for computing the Coeff-ideal. In step 8 the only remaining part of the function, i.e. $\varepsilon_1(-c;\dots)$ is being considered.

To make the algorithm 4.3 more readable, two further special cases have been omitted there: as soon as the maximal locus (on U_i) of the governing function consists of exactly one point, this point will be the center for the following blow-up; if on the other hand a candidate for a center is permissible and of codimension 1 in X (of the current auxilliary object), this will be the new center. In both cases the value of the further entries of the governing function is not determined.

Remark 4.3 *Although this seems strange at first glance, a step which needs to be very carefully treated from the practical point of view is the purely combinatorial step 4. If not done efficiently, this step can be very time-consuming.*

4.4 Further Enhancements

The three algorithms stated above basically allow us to determine the center of the following blow-up for a given basic object. In practice, however, there are a few more issues to be discussed: special situations, in which the way of describing the basic object can be optimized , and tricks to enhance the overall performance of the algorithm.

First of all, there is one step in the resolution process which we have not considered from the algorithmic point of view up to now: the blow-up itself. It is implemented as a preimage computation; more precisely, the ideal of the blown-up ambient space is determined as the preimage of $\mathcal{I}(W)$ under the map

$$\begin{array}{rcl} \phi : \mathbb{C}[x_1,\dots,x_n,y_0,\dots,y_{s-1}] & \longrightarrow & \mathbb{C}[x_1,\dots,x_n,t] \\ x_i & \longmapsto & x_i \\ y_i & \longmapsto & t \cdot g_i \end{array}$$

where the variables $x_1,\dots,x_n$ are the original ones, $y_0,\dots,y_{s-1}$ new variables and $g_0,\dots,g_{s-1} \in \mathbb{C}[x_1,\dots,x_n]$ a set of generators of the ideal of the center of the blow-up. All transforms are then computed in the usual way and the blown-up ambient space is then covered by the affine charts $D(y_i)$.

Obviously, the difficulty of the computation of the preimage depends very much on the generators g_i of the center and on the total number of variables involved. In particular, successive blowing-ups in smooth irreducible centers turn out to be by far less expensive than blowing-up at several smooth (disjoint) irreducible centers simultaneously. Therefore we always do a primary decomposition of the center, which has been determined by the algorithm, before proceding to the blow-up step; as the center is smooth, the components must be disjoint and the blow-ups at each of the components are independent of the other components of the center. In addition to this trick, checking whether there is a coordinate change such that after applying it the total number of occurring variables has dropped also speeds up the subsequent calculations significantly.

Another enhancement to the resolution process follows from the fact that not all s charts arising from a single blow-up contain new information. It may very well happen that in one or more charts we see neither a singular point of X nor an intersection of X and one of the elements of E or of two elements of E outside of the open subset already covered by the remaining charts. In this situation, we can drop this chart without doing any harm, because it only contributes information which is also present in the other charts. If we are only interested in a non-embedded resolution, we can further drop those charts in which the points which do not appear in the remaining charts do not meet X.

There is one last enhancement to the resolution process, which we use: at the very beginning the reduced part of the singular locus of the original basic object is determined. If it is smooth and irreducible, we will first use this as the center of the upcoming blow-up and enter the algorithm afterwards. As this only alters the very first step of the resolution process, it does not have any effect on the termination or correctness of the algorithm, but in certain examples it reduces the amount of calculations and charts significantly, like e.g. in the case of the Whitney umbrella, which immediately becomes non-singular after the first blow-up, or in the case of an isolated singularity of type U_{16} in which it cuts down the number of charts to be considered by a factor of 3.

4.5 Validation of the result

The previous considerations in this section dealt with the question of how to practically compute a resolution according to our algorithm in the most effective way. But there is also another important aspect to look at: the validation of the final result; more precisely, we discuss now how to check during the computation and a posteriori that the computed result is indeed a resolution of the given initial object. [17]

While computing the resolution, it is possible to check in each chart, whether the data in this chart contradicts the properties which a chart within the resolution process has to fulfil; in particular, it can be tested explicitly (for each respective chart)

- whether the ambient space W is non-empty and smooth
- whether the elements of E are smooth and normal crossing in this chart
- whether the center is permissible
- whether the center is contained in the union of the singular locus of X and the locus where X and E are not normal crossing

[17] These tests are also implemented and can be activated by supplying an additional argument to the resolution procedure.

For final charts, that is charts in which there is nothing left to do according to the algorithm, we can additionally test whether X is non-singular and normal crossing with E. These verifications can be done without reference to the construction.

But the tests, which we mentioned up to now, do not provide a complete set of tests for the implemented resolution algorithm, since all of these tests take place in a single chart. For further tests, it is necessary to identify the different exceptional divisors in each of the charts. A method for achieving this is discussed in the first subsection of the section 5. As soon as we know which element of E in a chart belongs to which exceptional divisor, we can then e.g. compare intersection loci of exceptional divisors (resp. of exceptional divisors and X) computed in different charts by moving from one chart to another one via blowing-downs and blowing-ups.

5 Applications

In this section, we present several applications of resolution of singularities: intersection matrix and genus of the exceptional curves in the case of non-embedded resolution of surface singularities, negative part of the spectrum of a hypersurface singularity and the Denef-Loeser zeta-function. But before we can start considering these more complex applications, we first have to find a good way to work with the resolved object, which is described in terms of the covering by the final charts; in particular, we need to determine whether a divisor in one chart and one in another chart both belong to the same exceptional divisor and even to the same component of it or not.

5.1 Identification of Exceptional Divisors

Given an embedded resolution of singularities as computed by the algorithm, we actually deal with a list of charts in which the transform and the exceptional divisors live. Therefore the first task is to find a way to identify points resp. subvarieties which appear in more than one chart; in particular we need to decide whether two given exceptional divisors living in two different charts actually belong to the same exceptional divisor of the global object. To this end, we will move through the tree of charts arising during the resolution process, first blowing-down from the first chart to the one in which the history of the two charts in question branched, and then blowing-up again to the other chart with which we want to compare (cf. figure 2).

As blow-ups are isomorphisms away from the center, this process of sucessively blowing-down and then blowing-up again does not cause any problems for points which do not lie on an exceptional divisor at all or only lie on exceptional divisors, which already exist in the chart at which the history of the considered charts branched. If, however, the point lies on an exceptional divisor which arises later, then blowing-down beyond the moment of birth of this divisor will inevitably lead to incorrect results, because this blow-up map is not

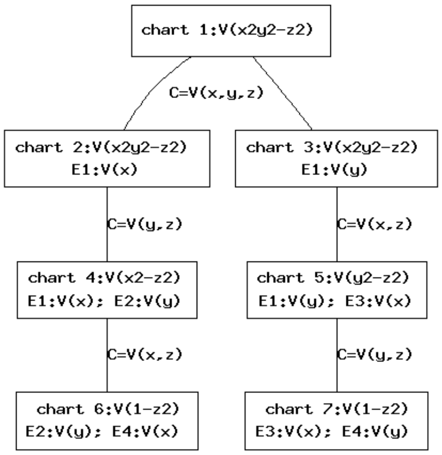

Figure 2: The tree of blow-ups in the resolution of the singularity $V(x^2y^2-z^2) \subset \mathbb{A}^3$. For simplicity of notation the variable names in all charts have been chosen to be x, y, z. To determine whether two exceptional divisors in two different charts actually belong to the same exceptional divisor, we need to move through the tree by first blowing down and then up again; for instance, the question, whether the divisors $V(x)$ in chart 6 and $V(y)$ in chart 7 belong to the same divisor, can only be answered by comparing the centers in charts 4 and 5. To this end, we have to move from chart 4 to chart one by blowing down twice and then proceed to chart 5 by blowing up twice.

an isomorphism. To avoid this problem, we need to represent the point on the exceptional divisor as the locus of intersection of the exceptional divisor with an auxilliary variety which is not contained in the exceptional divisor. More formally speaking, we use the following simple fact from commutative algebra:

Let $I \subset K[x_1,\ldots,x_n]$ be a prime ideal, $J \subset K[x_1,\ldots,x_n]$ another ideal such that $I+J$ is equidimensional and $ht(I) = ht(I+J) - r$ for some integer $0 < r < n$. Then there exist polynomials $p_1,\ldots,p_r \in I+J$ and a polynomial $f \in K[x_1,\ldots,x_n]$ such that

$$\sqrt{I+J} = \sqrt{(I+(p_1,\ldots,p_r)):f}.$$

In our situation, the ideal I is, of course, the ideal of the intersection of the exceptional divisors in which the point or subvariety $V(J)$ is contained. As any sufficiently general set of polynomials $p_1,\ldots,p_r \in J \setminus (I \cap J)$ leading to the correct height of $I+(p_1,\ldots,p_r)$ will do and as the only truely restricting condition on f is that it has to exclude all extra components of $I+(p_1,\ldots,p_r)$, we also have enough freedom of choice of the $p_1,\ldots,p_r, f$ to achieve that none of them is contained in any further exceptional divisor that might be in our way when blowing-down.

Having solved the problem of identifying points which exist in more than one chart, we can now determine which exceptional divisor in one chart coincides with which one in another chart by simply comparing the centers leading to these exceptional divisors. To this end, we start at the root of the tree of charts of the resolution and work our way up to the final charts. The criteria for identifying the centers are quite simple: first of all, the centers can not be the same, if the the corresponding values of the governing function do not agree, secondly, the centers cannot be the same if the exceptional divisors in which they are contained are not the same and, in the last step, the remaining candidates are compared explicitly by mapping them through the resolution tree as described above.

Remark 5.1 *In the case of non-embedded resolutions, the situation is a little more complicated. While in the embedded case, the exceptional divisors are always ($\mathbb{Q}$-)irreducible*[18]*, they might be reducible in the non-embedded case. To identify these components we therefore need to decompose the exceptional divisors before doing the comparison.*
Note that in this context, we are only able to deal with $\mathbb{Q}$-components in the implementation, since SINGULAR *computes over the rationals. The steps necessary to compute over the complex numbers will be mentioned in the upcoming section.*

5.2 Intersection Form of Exceptional Curves on a Blown-up Surface

Given an embedded resolution of a surface singularity, stored as a tree of charts, we would like to pass to a non-embedded resolution by dropping unnecessary blow-ups at the end of the branches of the tree of charts. To this end, we compute the list of exceptional divisors by identifying them in the different charts as described in the previous section 5.1. Starting at the final charts, we then move backwards through the resolution tree and cancel those blowing ups which are not necessary for the non-embedded resolution (see illustration 3 for an example).

Then we consider the intersection of the remaining exceptional divisors of the embedded resolution with the strict transform to obtain the exceptional locus of the non-embedded resolution. We can easily decompose these intersections into irreducible components over $\mathbb{Q}$,but these components may still be reducible over $\mathbb{C}$ and hence we need to achieve a descomposition over $\mathbb{C}$ to compute the intersection matrix of the exceptional divisors. The following theorem (cf. [8]) is the basis for the decomposition over $\mathbb{C}$:

[18]This is not a consequence of the algorithm itself, but of the two facts that the components of the center are always disjoint and that one of the enhancements discussed in the previous section was to do a primary decomposition of the center and blow-up in each of the components of the center separately

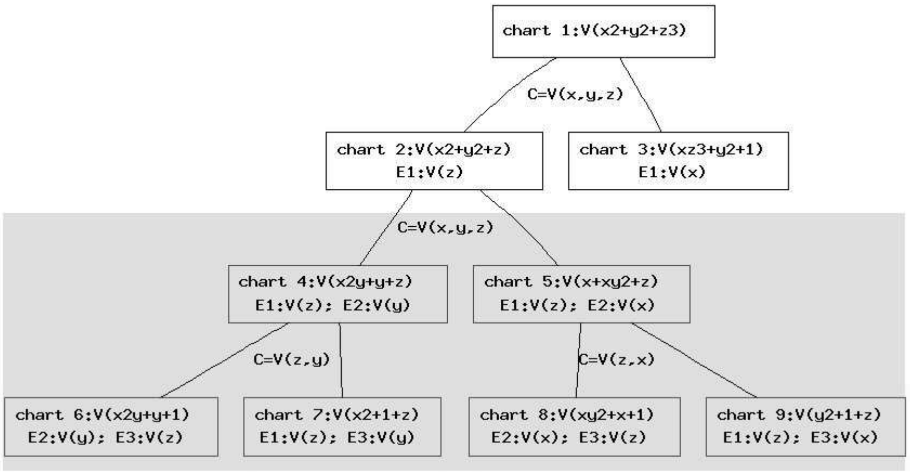

Figure 3: Tree of the embedded resolution process of an A_2 surface singularity. All charts which are marked by grey background arise from blow-ups which are only necessary in the embedded case, but not for a non-embedded resolution.

Theorem 5.2 (Gao/Ruppert) *Let $f \in \mathbb{Q}[x, y]$ be irreducible of bidegree (m, n). Let $G = \{g \in \mathbb{Q}[x, y] | (m-1, n) \geq deg(g), \exists h \in \mathbb{Q}[x, y], \frac{\partial(g/f)}{\partial y} = \frac{\partial(h/f)}{\partial x}\}$. The vector space G has the following properties*

(i) f is irreducible in $\mathbb{C}[x, y]$ if and only if $dim_{\mathbb{Q}}(G) = 1$.

(ii) $gG \subset \frac{\partial f}{\partial x} G$ mod f for all $g \in G$.

(iii) Let $g_1, \ldots, g_a \in G$ be a basis and $g \in G \smallsetminus \mathbb{Q}\frac{\partial f}{\partial x}, gg_i = \sum a_{ij} g_j \frac{\partial f}{\partial x}$ mod f. Let $\chi(t) = det(tE - (a_{ij}))$ be the characteristic polynomial. Then χ is irreducible in $\mathbb{Q}[t]$.

(iv) $f = \prod_{c \in \mathbb{C}, \chi(c)=0} gcd(f, g - c\frac{\partial f}{\partial x})$ is the decomposition of f into irreducible factors in $\mathbb{C}[x, y]$.

We use this theorem for the decomposition of curves in $\mathbb{C}^n$ which are irreducible over $\mathbb{Q}$ by means of the following corollary:

Corollary 5.3 *Let $I \subset \mathbb{Q}[x_1, \ldots, x_n], ht(I) = 1$, be a prime ideal. Then there exists an irreducible polynomial $\chi(t) \in \mathbb{Q}[t]$ such that the complex zeros of $\chi(t) = 0$ correspond to the associated prime ideals of $I\mathbb{C}[x_1, \ldots, x_n]$.*

Proof: Let such a curve be defined by a prime ideal $I \subset \mathbb{Q}[x_1, \ldots, x_n]$, $ht(I) = 1$. We may assume (after a generic linear coordinate change) that

$$\begin{aligned} I = \ (\ & a_1(x_n)x_1 - b_1(x_{n-1}, x_n), \ldots, \\ & a_{n-2}(x_n)x_{n-2} - b_{n-2}(x_{n-1}, x_n), b_{n-1}(x_{n-1}, x_n) \ ) \ : \ h \end{aligned}$$

with $b_{n-1} \in \mathbb{Q}[x_{n-1}, x_n]$ irreducible and a suitable $h \in \mathbb{Q}[x_n]$. Note that the set of polynomials $a_1(x_n)x_1 - b_1(x_{n-1}, x_n)$, ... , $a_{n-2}(x_n)x_{n-2} - b_{n-2}(x_{n-1}, x_n)$, $b_{n-1}(x_{n-1}, x_n)$ is a Groebner basis of $I\mathbb{Q}(x_n)[x_1, \ldots, x_{n-1}]$ with respect to the lexicographical ordering $x_1 > \ldots > x_{n-1}$ and $h = lcm(a_1, \ldots, a_{n-2})$.
If $b_{n-1} = d_1 \cdots d_k$ is the factorization of b_{n-1} over $\mathbb{C}[x_{n-1}, x_n]$ then

$$I = \cap((I, d_i) : h)$$

is the prime decomposition of I in $\mathbb{C}[x_1 \ldots x_n]$.
We have thus reduced our problem to the problem of factorization of a bivariate $\mathbb{Q}$-irreducible polynomial over the complex numbers; but the solution to this problem is given by 5.2 which provides the desired $\chi(t)$ by (iii). The correspondence of the zeros of $\chi(t)$ and the factors of the polynomial is statement (iv).

□

Over the field extension L of $\mathbb{Q}$ defined by χ, we can obtain at least one $\mathbb{C}$-irreducible factor. In general, this extension $L : \mathbb{Q}$ will not be Galois and b_{n-1} will not split completely. Theoretically, one would now pass to the corresponding Galois extension, but the practical computation thereof as well as explicit calculations in it tend to be very expensive. Therefore, we only compute one irreducible factor and additionally all solutions of $\chi(t) = 0$ by means of a numerical solver.

To illustrate this aspect, let us consider the factorization of the polynomial $x^3 + 2y^3$. In our implementation, the procedure `getMinpoly` computes the minimal polynomial described above and its numerical roots.

```
ring R=0,(x,y),dp;
poly p=x3-2y3;
getMinpoly(p);

[1]:
   poly p=t^3-2;
[2]:
   [1]:
      (-0.62996052494743658238360053+i*1.09112363597172140356007 26)
   [2]:
      (-0.62996052494743658238360053-i*1.0911236359717214035600726)
   [3]:
      1.2599210498948731647672106 1
[3]:
   3
```

If we factorize $x^3 + 2y^3$ over the field extension $\mathbb{Q}[t]/t^3 - 2$ we obtain two factors.

```
 ring T=(0,t),(x,y),dp;
 minpoly=t3-2;
 factorize(x3-2y3);

[1]:
   _[1]=1
   _[2]=x2+(t)*x*y+(t2)*y^2
   _[3]=x+(-t)*y
[2]:
   1,1,1
```

To obtain a complete factorization we need a Galois extension which is of higher degree. Therefore the factorization takes more time and so do all further calculations in this field (Just consider the large coefficients of y in our very simple example!).

```
 ring T=(0,t),(x,y),dp;
 minpoly=t6+3t5+6t4+11t3+12t2-3t+1;
 factorize(x3-2y3);

[1]:
   _[1]=1
   _[2]=x+(2/9t5+7/9t4+14/9t3+26/9t2+37/9t+2/9)*y
   _[3]=x+(1/9t5+2/9t4+4/9t3+4/9t2-1/9t-11/9)*y
   _[4]=x+(-1/3t5-t4-2t3-10/3t2-4t+1)*y
[2]:
   1,1,1,1
```

To identify the $\mathbb{C}$-components of an exceptional divisor E (irreducible over $\mathbb{Q}$) in a chart, we, therefore, store E, $\chi(t)$ and the respective numerical root of $\chi(t)$. Given these data, we can then proceed in the same way as for the identification of the $\mathbb{Q}$-components in 5.1. As soon as the exceptional divisors in the different charts are identified, we can directly compute the intersection numbers $E_i.E_j$ for all $i \neq j$.

The computation of the self-intersection numbers E_i^2 is done by the following well-known method:
Let $\pi : X \longrightarrow Y$ be a resolution of the surface Y and $E_1, \ldots, E_s$ the exeptional divisors and let $h : Y \longrightarrow \mathbb{C}$ be a non-trivial linear form. Then $\pi^*(h).E_i = 0$. Now we can write

$$\pi^*(h) = \sum_{i=1}^{s} c_i E_i + H,$$

where H denotes the strict transform, and obtain the equations

$$0 = \pi^*(h)E_i = \sum_{i=1}^{s} c_j E_j.E_i + H.E_i \quad \forall 1 \leq i \leq s$$

which provide us with the desired self-intersection numbers.

Example 5.4 *In our example at the beginning of this section (figure 3), the non-embedded resolution finishes after the first blow-up leading to the following situation in the charts:*

chart 2: *Forming the intersection of* $X = V(x^2 + y^2 + z)$ *and* $E = V(z)$ *leads to the* $\mathbb{Q}$*-irreducible curve* $E_1 = V(x^2 + y^2, z)$*, which obviously decomposes over* $\mathbb{C}$ *into the two curves* $E_{1,1} = V(x + iy, z)$ *and* $E_{1,2} = V(x - iy, z)$*. These meet in one point.*

chart 3: *In this chart, we see the same two curves* $E_{1,1}$ *and* $E_{1,2}$*, but we do not see the point in which they intersect.*

Therefore we are dealing with two exceptional divisors for which the intersection matrix is

$$\begin{pmatrix} * & 1 \\ 1 & * \end{pmatrix},$$

where the places for the (not yet computed) self-intersection numbers are filled by $*$*. To determine these, we choose some linear form* h*, e.g. given by the polynomial* x*. As the next step, we need to determine the divisor of* $\pi^*(h)$ *in terms of the* E_i *in each of the chart to make sure that we do not miss out any contribution.*

chart 2: $\pi^*(h) = 1 \cdot E_{1,1} + 1 \cdot E_{1,2} + V(x, y^2 + z)$

chart 3: $\pi^*(h) = 1 \cdot E_{1,1} + 1 \cdot E_{1,2} + V(1)$

This allows us to determine the self-intersection numbers as

$$\begin{aligned} 0 = \pi^*(h).E_{1,1} &= 1 \cdot E_{1,1}.E_{1,1} + 1 \cdot E_{1,2}.E_{1,1} + H.E_{1,1} \\ &= E_{1,1}^2 + 1 + 1 \\ 0 = \pi^*(h).E_{1,2} &= 1 \cdot E_{1,1}.E_{1,2} + 1 \cdot E_{1,2}.E_{1,2} + H.E_{1,1} \\ &= 1 + E_{1,2}^2 + 1 \end{aligned}$$

Therefore the complete intersection matrix is

$$\begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix}$$

5.3 Genus of Exceptional Curves on a Blown-up Surface

As soon as we have identified the $\mathbb{C}$-components of the exceptional divisors on the surface (as described in the preceding section), we are able to compute the genus of one $\mathbb{C}$-component. To this end, we first observe that the geometric genus is a birational invariant and therefore already determined by a single chart. Considering a $\mathbb{Q}$-irreducible exceptional divisor E and its decomposition $E = \cup_{i=1}^{a} E_i$ over $\mathbb{C}$ then we have

$$g(E) = ag(E_i) - a + 1 \quad \forall 1 \leq i \leq a.$$

Therefore the genus of the E_i can be computed from the genus of E, which, in turn, is already available in SINGULAR.

Example 5.5 *In the example of the previous section (figure 3), we choose chart 2 to determine the genus of $E_{1,1}$ and $E_{1,2}$. To this end, we compute the genus of E: $g(E) = -1$ and conclude by the above formula that both $E_{1,i}$ are of genus zero - which is exactly what we expected.*

Using SINGULAR, the computations of 5.4 and 5.5 look as follows:

```
LIB "zeta.lib";
ring R=0,(x,y,z),dp;
ideal I=x2+y2+z3;
list L=resolve(I);
intersectionDiv(L);
[1]:
   -2, 1
    1,-2
[2]:
    0,0
[3]:
   [1]:
      [1]:
          2,1,1
      [2]:
          3,1,1
```

5.4 Spectrum of a Hypersurface Singularity

In this section, we explain how to compute the negative part of the spectrum of an isolated hypersurface singularity using the resolution of the singularity. For a definition of the spectrum see e.g. [14],[15] Note that (denoting by μ the Milnor number) the spectrum $S = \{\alpha_1, \ldots, \alpha_\mu\}$ is a set of rational numbers in the open interval $(-1, n-1)$, where n the dimension of the ambient space. Let $f \in \mathbb{C}[x_1, \ldots, x_n]$ be a non-zero polynomial defining a hypersurface V with one isolated singularity at the origin and let $\pi : X \longrightarrow \mathbb{C}^n$ be an embedded resolution V. Let $E_i, 1 \leq i \leq s$ be the ($\mathbb{C}$-irreducible) exceptional divisors. For each $1 \leq j \leq s$, we denote by $N(E_j)$ the multiplicity of E_j in the divisor of $f \circ \pi$ and by $\nu(\omega, E_j) - 1$ the multiplicity of E_j in the divisor of $\pi^*(\omega)$, where ω is a holomorphic n-form in $\mathbb{C}^n$. In particular, $\nu(E_j) = \nu(dx_1 \wedge \ldots \wedge dx_n, E_j)$.
For each holomorphic n-form ω in $\mathbb{C}^n$ the geometric weight $g(\omega)$ with respect to the resolution is defined by

$$g(\omega) = min_{i \in I}\{\nu(\omega, E_i)/N(E_i)\} \ .$$

The following lemma is the basis for the computation of the negative part of the spectrum:

Lemma 5.6 (Varchenko) *A non-positive rational number s is in the spectrum S if and only if there exsist a holomorphic n-form ω in $\mathbb{C}^n$ such that $s = g(\omega)-1$.*

Let us consider the following example:

```
LIB"zeta.lib";
ring R=0,(x,y,z),dp;
ideal I=(xz+y2)*(xz+y2+x2)+z5;
list L=resolve(I);

spectralNeg(L);

[1]:
   -11/40
[2]:
   -3/40
[3]:
   -1/20
[4]:
   -1/40
```

Remark 5.7 *In practice, all computations can be done over $\mathbb{Q}$:*
If D is a $\mathbb{Q}$- irreducible divisor in the resolution then D is a **disjoint** *union of some of the E_i from this resolution and $\nu(\omega, E_i) = \nu(\omega, D)$ resp.$N(E_i) = N(D)$.*

5.5 ζ-Function

In this section, we will describe how to compute the Denef-Loeser ζ - function for a given hypersurface singularity [19]. The Denef-Loeser ζ - function $Z_{top}^{(d)}(f, s)$ is defined as follows:
Let $f \in \mathbb{C}[x_1, \ldots, x_n]$ be a non-zero polynomial defining a hypersurface V and let $\pi : X \longrightarrow \mathbb{C}^n$ be an embedded resolution of V. Denote by E_i , $i \in I$, the irreducible components of the divisor $\pi^{-1}(f^{-1}(0))$. For each subset $J \subset I$ we define

$$E_J := \cap_{j \in J} E_j \text{ and } E_J^* := E_J \smallsetminus \cup_{j \notin J} E_{J \cup \{j\}} \ .$$

For each $j \in I$ let $N(E_j)$ and $\nu(E_j)$ be defined as in the previous section. The global Denef-Loeser ζ - function of f is

$$Z_{top}^{(d)}(f, s) := \sum_{\substack{J \subset I \text{s.th.} \\ d | N(E_j) \forall j \in J}} \chi(E_J^*) \prod_{j \in J} (\nu(E_j) + N(E_j)s)^{-1} \in \mathbb{Q}(s).$$

[19] In practice, only the case of dimension 2 is implemented

Intersecting the E_J^* with the preimage of zero in the above formula leads to the local Denef-Loeser zeta function

$$Z_{top,0}^{(d)}(f,s) := \sum_{\substack{J\subset I \text{s.th.} \\ d|N(E_j)\forall j\in J}} \chi(E_J^* \cap \pi^{-1}(0)) \prod_{j\in J} (\nu(E_j) + N(E_j)s)^{-1} \in \mathbb{Q}(s).$$

Lemma 5.8 *Let D_l , $l \in L$, be the $\mathbb{Q}$-irreducible components of the divisor $\pi^{-1}(f^{-1}(0))$. For each subset $J \subset L$ define D_J and D_J^* as above. Then*

$$Z_{top}^{(d)}(f,s) := \sum_{\substack{J\subset L \text{s.th.} \\ d|N(D_j)\forall j\in J}} \chi(D_J^*) \prod_{j\in J} (\nu(D_j) + N(D_j)s)^{-1} \in \mathbb{Q}(s), resp.$$

$$Z_{top,0}^{(d)}(f,s) := \sum_{\substack{J\subset I \text{s.th.} \\ d|N(D_j)\forall j\in J}} \chi(D_J^* \cap \pi^{-1}(0)) \prod_{j\in J} (\nu(D_j) + N(D_j)s)^{-1} \in \mathbb{Q}(s).$$

Proof: The D_l are smooth and therefore disjoint unions of some of the E_j. This implies that for $M = \{l_1,\ldots,l_q\} \subset L$ we obtain $\chi(D_M^*) = \sum \chi(E_J^*)$, the sum is on all subsets $J = \{j_1,\ldots,j_q\} \subset I$ such that E_{j_i} is a component of D_{l_i}. Moreover, it is easy to see that for a component E_j of D_l always $N(E_j) = N(D_l)$ and $\nu(E_j) = \nu(D_l)$.

$\square$

In the case of surfaces (n=3) the Euler characteristic $\chi(D_J^*)$ of the D_J^* can be computed as follows:

(i) $\chi(D_j^*) = 0$ for each $\mathbb{Q}$-component of the strict transform

(ii) For an exceptional divisor D_j the Euler characteristic $\chi(D_j)$ can be computed via the resolution tree using the following facts: Let $\sigma : Y \longrightarrow Z$ be a single blow-up step with center C in the resolution tree leading to the exceptional divisor E and let D be a divisor in Z intersecting C transversally with strict transform $\tilde{D}$.

   (a) $\chi(E) = k \cdot \chi(\mathbb{P}^2) = 3k$ if C is a $\mathbb{Q}$-irreducible zero-dimensional center corresponding to k points over $\mathbb{C}$.

   (b) $\chi(E) = \chi(\mathbb{P}^1 \times C) = 4 - 4g(C)$ if C is a curve.

   (c) $\chi(\tilde{D}) - \chi(E \cap \tilde{D}) = \chi(D) - \chi(C \cap D)$

(iii) If $J = \{i,j,k\}$ then $\chi(D_J^*) = \#(D_i \cap D_j \cap D_k)$.

(iv) $\chi(D_i \cap D_j) = 2 - 2g(D_i \cap D_j)$.

(v) If $J = \{i,j\}$ then $\chi(D_J^*) = \chi(D_i \cap D_j) - \sum_{k,k\neq i,j} \#(D_i \cap D_j \cap D_k)$.

(vi) $\chi(D_i^*) = \chi(D_i) - \sum_{i\neq j} \chi(D_i \cap D_j) + \sum_{j,k\neq i} \#(D_i \cap D_j \cap D_k)$.

Example 5.9 *Let us consider the same example as before, an A_2 surface singularity:*
As already illustrated in figure 3, the tree of blow-ups of the resolution process consists of 9 charts and we have 2 exceptional divisors arising from blow-ups whose center is a point and 1 from a blow-up whose center is 1-dimensional. The strict transform of the original V is irreducible and will be denoted by E_4. Since the surface only has one singular point at the origin, the global and the local Denef-Loeser zeta function coincide in this case.

As can easily be checked by direct calculation, we have

J	$\chi(E_J^*)$	J	$\chi(E_J^*)$	J	$\chi(E_J^*)$
1	0	1,2	0	1,2,3	0
2	1	1,3	0	1,2,4	0
3	0	1,4	2	1,3,4	2
4	0	2,3	2	2,3,4	0
		2,4	0		
		3,4	0		

For the multiplicities, we obtain

	E_1	E_2	E_3	E_4
N	2	3	6	1
ν	3	5	8	1

Hence, we obtain the Denef-Loeser zeta-function

$$\begin{aligned} \frac{1}{3s+5} &+ \frac{2}{(2s+3)(s+1)} + \frac{2}{(3s+5)(6s+8)} + \frac{2}{(2s+3)(6s+8)(s+1)} \\ &= \frac{s+4}{(3s+4)(s+1)} \end{aligned}$$

Using SINGULAR, the computation looks as follows:

```
LIB"zeta.lib";
ring R=0,(x,y,z),dp;
ideal I=x2+y2+z3;
list re=resolve(I);
zetaDL(re,1);
[1]:
(s+4)/(3s2+7s+4)
```

A Implementation and Timings

All algorithms described in this article have been implemented and are available as part of the SINGULAR software package (versions 2.2 and higher). All implementations have been carried out in the programming language of SINGULAR.

As this language is an interpreted one, the reader might hope for some speed-up by implementing the algorithms as dynamic modules written in C/C++; but this would not have the great impact on the timings, because a number of tools, such as radical, primary decomposition, singular locus etc., which are used internally in the resolution algorithm, are themselves implemented in the language of SINGULAR and not in C/C++.

A significant speed-up, however, may be achieved by using the fact that (due to the use of charts for representing the basic object) the structure of the resolution algorithm itself is parallel in a natural way. To see this, the most important observation is that the center is determined by finding the maximal value of the governing function in each chart. This is a task which can easily be performed parallely, i.e. simultaneously in several charts. On the other hand, this is the most time consuming single step of the resolution process and hence parallelizing it speeds up the whole process.[20]

The following table shows timings for the resolution algorithm in the embedded and in the non-embedded case[21]; most examples are surfaces, the last one is of dimension 5.The timings were taken on an Intel Celeron 2 GHz, 256 MB RAM:

Example		non-embedded		embedded	
	equations	mem. in MB	time in s	mem. in MB	time in s
A_{10}	$x^{11} + y^2 - z^2$	1	3	1	2
A_{20}	$x^{21} + y^2 - z^2$	2	5	1	4
D_{11}	$x^2y + y^{10} - z^2$ 4	23		4	17
E_6	$x^3 + y^4 - z^2$	1	4	1	3
non-isol.	$z^2 - x^3y^3$	-	-	1	6
W_{17}	$y^7 + xy^5 + x^4 + z^2$	4	10	4	10
Duco[22]	$x^2y^2z^2 + x^7 + y^8 + z^8$	4	19	3	17
W_8^*	$z^2 - xy^2$, $yz - x^3$, $y^3 - x^2z$	-	-	1	2
$E_8^\#$	$x_1x_6 - x_2x_3$, $x_1^4 + x_1x_2^5 - x_3x_5$, $x_1^3x_2 + x_2^6 - x_5x_6$	-	-	3	17

Now we consider the timings for the main steps inside the resolution algorithm separately in the example $z^2 - x^3y^3$. The listed times for the blow-ups and for determining the center are the sums of the times for all charts contributing to the step:

[20] The implementation of the parallelization has not yet reached a final stage and hence we do not have final timings for it.

[21] Currently only implemented in some special situations

[22] Example suggested for testing the software by Duco van Straten.

	invariant at center	#charts	time for center in ms	time for blow-up in ms
1	(2,0,0;6,0)	1	360	70
2	(2,0,0;3,0)	2	460	100
3	(2,0,(-1;2);1,1)	2	450	130
4	(2,0,(-1;1);1,1)	2	430	140
5	(1,2,0;2,0)	2	150	210
6	(1,2,(-1;1);1,1)	2	140	330
7	(1,1,(-1;1);1,1)	4	830	390
8	(1,1,$-\infty$)	4	810	440
9	final charts	18	660	-
	total	37	4290	1810

In the following table, there are some timings (in seconds) for those applications which have been presented in section 5:

Example	identify divisors	intersection matrix and genus of the E_i	Denef-Loeser zeta-funct.
A_{10}	2	11	15
A_{20}	5	26	41
E_6	3	13	24
W_{17}	14	29	121
Duco	68	105	694

For visualizing the tree of the resolution process and for drawing the Dynkin diagrams, the external program `dot` is used which is available as part of the `graphviz` software package from

`http://www.research.att.com/sw/tools/graphviz/`

This package is not part of the Singular software. Communication between Singular and `dot` is managed by the Singular procedures in the library `resgraph.lib`.

B The Local Case

In this section we describe how to construct in an efficient way an embedded resolution of a germ $(X,0) \subset (\mathbb{C}^n,0)$. The idea is to find a suitable representative such that its singular locus has only components through the point 0 and use the global resolution for this representative.

Let $X \subset \mathbb{C}^n$ be a representative of $(X,0)$, defined by the ideal $I \subset \mathbb{C}[x_1,\dots,x_n]$ and let the point 0 be in the singular locus of X. The singular locus of X may also have components not containing 0; they do not need to be resolved. To avoid resolving them, we choose a suitable open subset of X containing 0 such that the components of the singular locus not passing through 0 do not meet this open subset:
Using primary decomposition, we can describe the singular locus $Sing(X)$ of

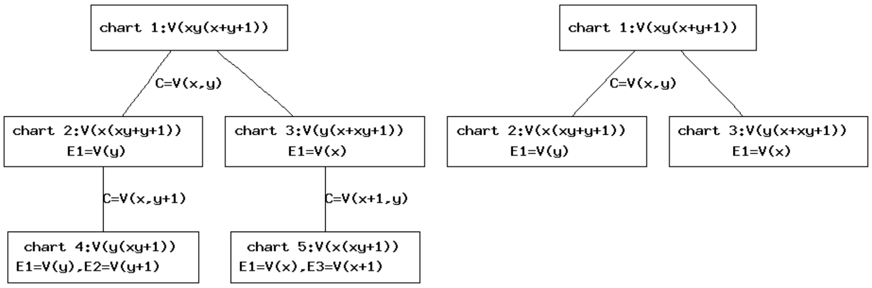

Figure 4: Two trees of resolution processes for an affine variety consisting of three lines in the plain (in general position). The left tree shows the process of resolution all three singularities of type A_1, whereas the right one shows the 'local' resolution of the singularity, only resolving the A_1 singularity at $V(x, y, z)$.

X as $\mathcal{I}(Sing(X)) = J_0 \cap J_1$ such that $J_0 \subset (x_1, \ldots, x_n)$ and $J_1 \not\subset (x_1, \ldots, x_n)$. Choose an element $h \in J_1$ which is not contained in $(x_1, \ldots, x_n)$. Then the open subset $X_h := X \cap D(h)$ of X defined by h is also a representative of $(X, 0)$. Now all components of the singular locus of this representative contain the point 0.

On the other hand, the open set X_h of X is isomorphic to the affine variety $Y \subset \mathbb{C}^{n+1}$ defined by the ideal $I\mathbb{C}[x_1, \ldots, x_n, t] + (th-1)\mathbb{C}[x_1, \ldots, x_n, t]$; that is the germ $(Y, 0) \subset \mathbb{C}^{n+1}$ is isomorphic to the germ $(X, 0) \subset (\mathbb{C}^n, 0)$.

We can use the global resolution for Y and avoid resolving singularities of X not being on a component of the singular locus through the point 0.
In practice, this approach to compute a global resolution of Y can be more complicated then to compute a global resolution of X. The reason is that the the blowing up of $(Y, 0) \subset \mathbb{C}^{n+1}$, which involves Gröbner basis computations in a ring with a higher number of variables and also deals with a higher number of equations, is usually more complicated. To avoid this drawback, we use the following considerations:

$$\begin{tikzcd}
\pi(C) \arrow[r, hook] & X_h \arrow[r, hook] & D(h) \arrow[r, hook] & \mathbb{C}^n \\
C \arrow[r, hook] \arrow[u, "\cong"] & Y \arrow[r, hook] \arrow[u, "\cong"] & V(th-1) \arrow[r, hook] \arrow[u, "\cong"] & \mathbb{C}^{n+1} \arrow[u, "\pi"]
\end{tikzcd}$$

In this diagram, the map π is defined by $\pi(x, t) = x$. A permissible center C for Y defines a permissible center $\overline{\pi(C)}$ for X; the blow-up of Y in C is isomorphic to the open subset defined by h in the blow-up of X in $\overline{\pi(C)}$. Therefore it is possible to use Y only for determining the center and then blow-up X in the closure of the projection of this center.

C Examples: Resolutions of Some Singularities

In this section, we list the SINGULAR sessions for the examples which were mentioned in the preceding sections. In all sessions, we assume that the required libraries are already loaded.

C.1 Surface Singularity A_{10}

As the first example, we have chosen an A_k surface singularity, since the results for these singularities are well known and allow the reader to compare the computed results in a convenient way.

```
ring R=0,(x,y,z),dp;
ideal I=x11+y2-z2;              // an A10 surface singularity
list L=resolve(I);              // compute its resolution
list coll=collectDiv(L);        // identify the divisors
ResTree(L,coll[1]);
```

The last command `ResTree` produces an illustration of the tree of blow-ups and presents the output which is shown in figure 5 in a separate window.

As this is the first example, which we are listing here, we include the rather lengthy output of the computation of the intersection matrix of the exceptional curves, only truncating it as soon as the structure has become clear.

```
                                // intersection matrix and genus of
                                // the exceptional curves on the
list iD=intersectionDiv(L);     // blown-up surface
iD;
[1]:
   -2,0,1,0,0,0,0,0,0,0,
   0,-2,0,1,0,0,0,0,0,0,
   1,0,-2,0,1,0,0,0,0,0,
   0,1,0,-2,0,1,0,0,0,0,
   0,0,1,0,-2,0,1,0,0,0,
   0,0,0,1,0,-2,0,1,0,0,
   0,0,0,0,1,0,-2,0,1,0,
   0,0,0,0,0,1,0,-2,0,1,
   0,0,0,0,0,0,1,0,-2,1,
   0,0,0,0,0,0,0,1,1,-2
[2]:
   0,0,0,0,0,0,0,0,0,0
[3]:
   [1]:
      [1]:
         2,1,1
      [2]:
```

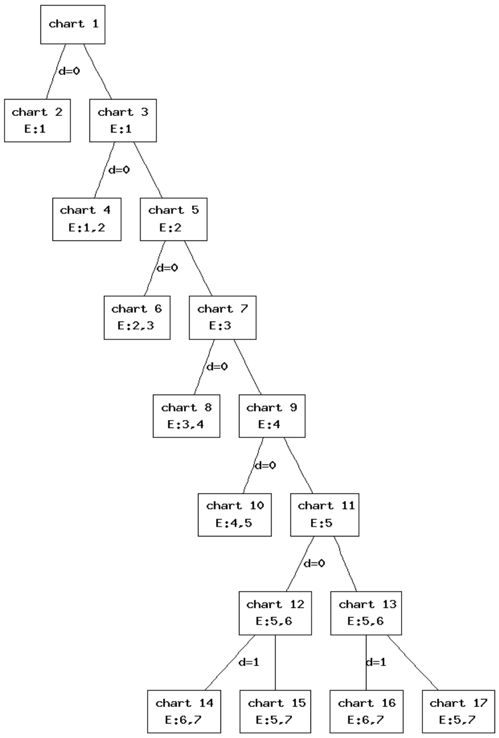

Figure 5: Tree of charts of the resolution of an A_{10} surface singularity.

```
        4,1,1
    [2]:
       [1]:
          2,1,2
       [2]:
          4,1,2
  .
  .
  .
    [10]:
       [1]:
          10,5,2
       [2]:
          11,5,1
                                // draw illustration of the
interDiv(iD[1]);                // intersection matrix
```

In the preceding output, the first list entry is the intersection matrix, the second one lists the genus of the exceptional curves and the last one specifies in which charts and as which component of which entry of the list of exceptional divisors each curve occurs. The illustration of the intersection matrix produced by the command `interDiv` appears in a separate window; here it is shown in figure 6.

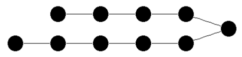

Figure 6: Coxeter-Dynkin-diagram of an A_{10} surface singularity. (The structure of the picture is slightly unusual because it has been generated automatically; but it is obviously the diagram, which we would expect.)

```
                            // Denef-Loeser zeta function
                            // isolated singularity, hence global
                            // and local coincide
zetaDL(L,1);                // with d=1
[1]:
   (s+12)/(11s2+23s+12)
zetaDL(L,2);                // with d=2
[1]:
   0
```

C.2 Surface Singularity E_6

```
ring R=0,(x,y,z),dp;
ideal I=x3+y4-z2;                // an E6 surface singularity
list L=resolve(I);               // compute its resolution
list coll=collectDiv(L);         // identify the divisors
ResTree(L,coll[1]);
list iD=intersectionDiv(L);      // intersection matrix and genus
InterDiv(iD[1]);
zetaDL(L,1);
[1]:
   (6s+13)/(12s2+25s+13)
```

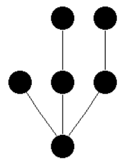

Figure 7: Coxeter-Dynkin diagram of an E_6 surface singularity as computed by the Singular library.

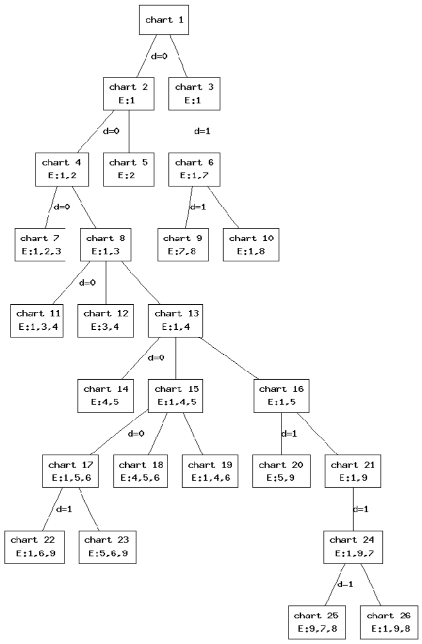

Figure 8: Tree of blow-ups for the E_6 surface singularity.

The output of the commands `ResTree` and `interDiv` is shown in figure 8 resp. 7. As the latter already illustrates the intersection matrix, we have not printed the output of `intersectionDiv` here.

C.3 Duco's Example

```
ring R=0,(x,y,z),dp;
ideal I=x2y2z2+x7+y8+z8;        // again a surface singularity
list L=resolve(I);              // compute the resolution
list coll=collectDiv(L);        // identify the divisors
                                // show tree of charts -- see
ResTree(L,coll[1]);             //        separate figure
list iD=intersectionDiv(L);     // intersection matrix and genus
[1]:
   -5,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,
   1,-5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,
   0,0,-10,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,1,-2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,1,0,-2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,1,0,0,-2,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,1,0,0,0,-2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

```
   0,0,0,1,0,0,0,-2,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,1,0,0,0,-2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,1,0,0,-2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,1,0,0,0,0,-2,0,0,0,1,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0,1,0,0,-2,0,0,0,1,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,1,0,0,0,0,-2,0,0,0,1,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,0,1,0,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,0,1,0,0,0,0,0,0,
   0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,0,0,1,0,0,0,0,
   0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,1,0,0,0,0,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,0,0,1,0,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,-2,0,0,1,0,0,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,-2,0,0,0,1,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,-2,0,0,1,0,
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,-2,0,0,1,
   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,-2,0,1,
   1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,-2,0,
   0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,-2
[2]:
   0,0,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
[3]:.
    .
    .
spectralNeg(L);                // negative spectral numbers
[1]:
   -1/2
[2]:
   -3/8
[3]:
   -5/14
[4]:
   -1/4
[5]:
   -13/56
[6]:
   -3/14
[7]:
   -1/8
[8]:
   -3/28
[9]:
   -5/56
[10]:
   -1/14
                               // isolated singularity ==>
zetaDL(L,1);                   // local and global zeta fct. coincide
```

```
[1]:
   (1120s3+1413s2+618s+98)/(784s4+1960s3+1764s2+686s+98)
```

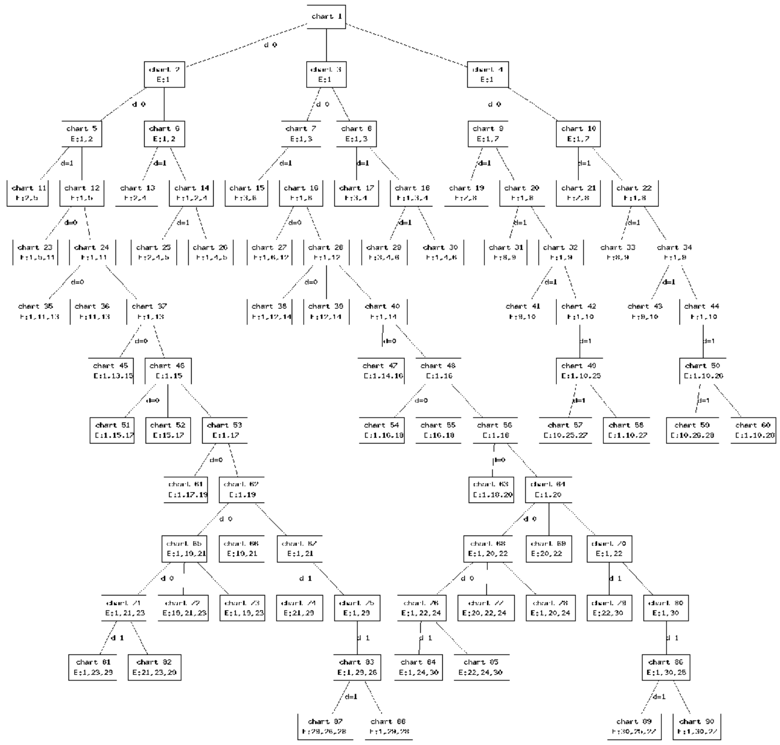

Figure 9: Tree of charts of example C.3. An interesting fact to observe here is the symmetry between two branches of the tree which is due to the invariance of the original equation under permutation of y and z.

C.4 A Surface Singularity Leading to an Exceptional Curve of Genus 1

```
ring R=0,(x,y,z),dp;
ideal I=x6+y3+z3;              // the surface singularity
list L=resolve(I);             // compute the resolution
list coll=collectDiv(L);       // identify the divisors
                               // show tree of charts -- see
ResTree(L,coll[1]);            //        separate figure
list iD=intersectionDiv(L);    // intersection matrix and genus
iD;
[1]:
   -2,0,0,1,
```

```
   0,-2,0,1,
   0,0,-2,1,
   1,1,1,-3
[2]:
   0,0,0,1
[3]:
   [1]:
      [1]:
         2,1,1
      [2]:
         4,1,1
   [2]:
      [1]:
         2,1,2
      [2]:
         4,1,2
   [3]:
      [1]:
         4,2,1
      [2]:
         5,2,1
```

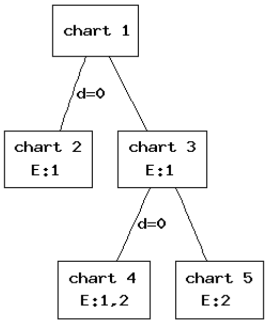

Figure 10: Tree of blow-ups of example C.4.

Comparing the tree of blow-ups (figure 10) with the intersection matrix, we immediately see that there have only been two blow-ups, but we are dealing with 4 exceptional curves on the resolved surface. As the last divisor is the only one with genus one, a first guess may be that the first exceptional curve consisted of three $\mathbb{C}$-components. To check this guess, we change to chart 4, in which both exceptional divisors are present:

```
                                    // total list of charts is L[2]
def S4=L[2][4];                     // 4th chart is L[2][4]
```

```
setring S4;                          // change to the ring
BO;                                  // look at the basic object
[1]:
   _[1]=0
[2]:
   _[1]=y(0)^3+y(2)^3+1
[3]:
   1
[4]:
   [1]:
      _[1]=y(2)
   [2]:
      _[1]=x(2)
[5]:
   _[1]=x(2)*y(2)
   _[2]=x(2)^2*y(0)*y(2)
   _[3]=x(2)^2*y(2)
[6]:
   0,0
[7]:
   2
[8]:
   _[1,1]=0
   _[1,2]=1
   _[2,1]=0
   _[2,2]=0
[9]:
   0
```

The strict transform of the surface is specified by the ideal `BO[2]`, i.e. it is $V(y(0)^3+y(2)^3+1)$, the exceptional divisors are listed in `BO[4]`. Therefore the first exceptional divisor induces the exceptional curve $V(y(0)^3+1, y(2))$ which is obviously the disjoint union of three curves. The second exceptional curve is $V(y(0)^3+y(2)^3+1, x(2))$, an elliptic curve.

```
                                // isolated singularity ==>
zetaDL(L,1);                    // local and global zeta fct. coincide
[1]:
   (2s+5)/(6s2+11s+5)
```

C.5 A 4-dimensional variety with embedding dimension 6

```
ring R = 0,(x(1..6)),dp;        // just to show that we can not
                                // only do curves surfaces and 3-folds
ideal J=x(1)*x(6)-x(2)*x(3),
        x(1)*(x(1)^3+x(2)^5)-x(3)*x(5),
```

```
        x(2)*(x(1)^3+x(2)^5)-x(5)*x(6);
dim(std(J));                        // check the dimension
4
list L=resolve(J);                 // compute the resolution
list coll=collectDiv(L);           // identify the divisors
ResTree(L,coll[1]);                // see figure below
```

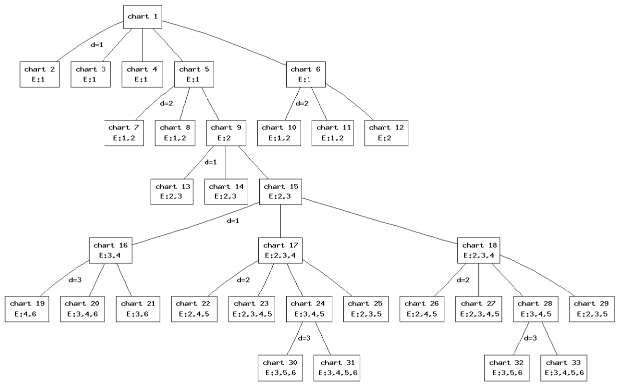

Figure 11: Tree of charts of example C.5.

C.6 A non-isolated surface singularity: Local and Global Zeta Function

```
ring R=0,(x,y,z),dp;
ideal I=xy*(x+y+1);              // three planes
minAssGTZ(slocus(I));            // singular locus: 3 disjoint lines
[1]:
   _[1]=y+1
   _[2]=x
[2]:
   _[1]=y
   _[2]=x+1
[3]:
   _[1]=y
   _[2]=x
list L=resolve(I);
zetaDL(L,1);                     // the global Denef-Loeser zeta fct.
6/(s2+2s+1)
zetaDL(L,1,"lokal");             // the local Denef-Loeser zeta fct.
1/(s2+2s+1)
```

D Data Structures for the Resolution Library in SINGULAR

As SINGULAR only supports elementary data types like integers, polynomials, ideals etc. and the general purpose data type list, it is not possible to specify user defined structure data types. On the other hand, considering basic objects, exceptional divisors appearing in more than one chart and, of course, representing the whole resolution requires storing of structured data. This has been implemented in the resolution library by means of lists whose entries satisfy certain conditions. Since these extra conditions are used everywhere in the code when dealing with these lists, we specify then in this appendix to allow easier interpretation of the output of the procedures.

D.1 Data Structure for the Resolution Process

Basic Object

In the resolution library, a basic object $\mathcal{B} = (W, X, b, E, v)$ is described by means of an affine covering. Each of the affine charts corresponds to a ring in which there is a list of 'type' BO, that is a variable of type list bearing the name $\mathcal{B}$ which is subject to the following condition on the structure of the list:

1. `ideal`, describing W
2. `ideal`, describing X
3. `intvec`, the i-th entry of this vector of integers is the integer b of the $(i-1)$st auxilliary object of $\mathcal{B}$
4. `list` of 'type' exceptional divisors describing E
5. `ideal`, describing the birational morphism leading from the original basic object to this one - the i-th entry of the ideal is the image of the i-th variable from the original ring
6. `intvec`, internal data describing certain special intersection properties of the elements of the set of exceptional divisors
7. `intvec`, the sum over the first i entries of this vector of integers is the index of the last element of the list of exceptional divisors which will not appear in the i-th auxilliary object
8. `intmat`, describing the intersections of the elements of E in this chart by $M_{ij} = 1$ iff $E_i \cap E_j \neq \emptyset$
9. `intvec`, the i-th entry of this vector of integers is the integer n of the $(i-1)$st auxilliary object, that is the maximal number of exceptional divisors intersecting X simultaneously in this chart, whose index is at most the sum over the first i entries of the item 7 in the basic object list
10. `list` of 'type' saved centers

Exceptional Divisors

The 4-th entry of the basic object list is of 'type' exceptional divisor; it is a list of `ideal`s, each of which describes a hypersurface in W. Moreover, the entries are assumed to appear in the list in the order in which they have arisen in the resolution process.

Saved Center

Each entry of a list of 'type' saved centers is itself a list of the following form:

1. `ideal`, describing a component of the center
2. `intvec`, coinciding with entry 7 of the basic object list at the time when the primary decomposition of the center took place
3. `intvec`, coinciding with entry 3 of the basic object list at the time when the primary decomposition of the center took place
4. `intvec`, coinciding with entry 9 of the basic object list at the time when the primary decomposition of the center took place

Chart

A single chart in the resolution is represented by a ring containing some or all of the following variables whose names and 'types' are fixed:

BO `list` of type basic object

cent `ideal`, describing the upcoming center determined by the algorithm

path `module` used as a matrix, the i-th column of this matrix contains the index of the appropriate chart of the $(i-1)$st step in the resolution history of the current chart as first entry and the index of the chart of the blow-up leading to the next step in the history of the current chart.

lastMap `ideal`, describing the very last of the sequence of blow-ups leading to the current chart

invCenter `list` of 'type' CenterData

From this list, the variable BO is always present, path and lastMap are only defined if there have been previous blow-ups. If a center has been computed, it is stored in cent, and if cent is defined and the procedure prepareInv has previously been called, invCenter is defined.

CenterData

CenterData is a `list` describing the center for the upcoming blow-up in the following way:

1. `ideal` describing the center
2. `intvec` value of governing function at the center, that is $(b_1, n_1, -c_1; b_2, \ldots)$ (without specifying the index vector $i_1, \ldots, i_{c_j}$)
3. `intvec` marking the indices of the elements of E which contain the center by 1

Resolution of Singularities

A resolution of singularities is represented by a `list` containing two lists of charts (in the sense of the 'type' chart defined previously); the first list contains only the final charts, while the second one contains all charts.

D.2 Data Structures for special Applications

AbstractResolution (Non-embedded Resolution)

The implemented resolution algorithm computes an embedded resolution, from which we may produce a non-embedded one by simply stopping earlier as we saw in 5. Therefore the AbstractResolution contains the embedded resolution data combined with the information which charts are to be dropped.

1. `intvec` marking by a value of 1 the indices of the charts which are final charts in the non-embedded case
2. `intvec` marking by a value of 1 the indices of the charts which are children of final charts in the non-embedded case, i.e. which should not be present in the non-embedded case
3. `list` of 'type' resolution, describing the corresponding embedded resolution

DivisorList (Intersection Matrix and Genus)

A DivisorList is a `list` which contains the information about the identification of exceptional divisors in different charts in two different ways:
The first entry is of type `intmat`; this matrix contains as entry in the i-th row and j-th column the index of the exceptional divisor appearing as entry j in the 4-th entry of the basic object in chart i.
The second entry of the list is itself again a `list` whose i-th entry corresponds to the i-th exceptional divisor respectively whose last entry corresponds to the strict transform in the final charts. Each of these is, in turn, a `list` of `intvec`; an integer vector (i, j) corresponds to the j-th entry in the 4-th entry of BO

in the i-th chart, an integer vector $(i, 0)$ indicates that the strict transform appears in the final chart i and an integer vector (i, j, k)[23] implies it is the k-th component of the exceptional divisor corresponding to (i, j).

DecomposedE (Intersection Matrix and Genus)

A DecomposedE is a `list` in which the i-th entry corresponds to the i-th entry of BO in the same chart. Each of these entries of DecomposedE is itself a `list` whose entries correspond to a $\mathbb{Q}$-irreducible component of the divisor[24]. For each $\mathbb{Q}$- irreducible component the entry is again a list of the following structure:

1. `ideal`, describing the $\mathbb{Q}$-component
2. `int`, number of $\mathbb{C}$-irreducible components of this $\mathbb{Q}$-component
3. `int`, specifying n such that the component is an n-fold curve on the surface; in other words, the order of contact of the surface with the exceptional divisor (in a point of this component) considered as embedded in W
4. `intmat`, intersection matrix of the $\mathbb{C}$ -irreducible components
5. `string`, specifying the SINGULAR command for defining the minimal polynomial of a field extension of $\mathbb{Q}$ in which the $\mathbb{Q}$-irreducible component becomes reducible
6. `string`, specifying a general factor of the component by use of the parameter of the field extension
7. `list`, containing the numerical zeros of the minimal polynomial as `strings`

EulerList (Topological Zeta Function)

For the computation of the topological Zeta function, it is necessary to determine the Euler characteristic of the exceptional divisors and of the intersection of two resp. three exceptional divisors. To store these data, the 'type' EulerList is used which is basically a `list` containing three entries, the list of Euler characteristics of $E_i \cap E_j \cap E_k$, the one of Euler characteristics of $E_i \cap E_j$ and the one for the E_i. Each entry of these lists consists of a `list` whose first entry is an intvec specifying the indices involved and whose second entry is the Euler characterstic, an `int`.

[23] This only occurs in the non-embedded case after a decomposition of the exceptional divisors. In this case there is a list of type DecomposedE present in each of the final charts.

[24] Usually, this data type only occurs in the non-embedded case, where the exceptional divisors are of course specified by the ideal $BO[2] + BO[4][i]$ in the current chart.

References

[1] Abramovich,D., de Jong,A.J.:*Smoothness, Semi-Stability and Toroidal Geometry*, J.Alg.Geom. **6** (1997), pp.789–801

[2] Bierstone,E., Milman,P.:*Canonical Desingularization in Characteristic Zero by Blowing up the Maximum Strata of a Local Invariant*, Invent.Math. **128** (1997), pp. 207–302

[3] Bierstone,E., Milman,P.:*Desingularization Algorithms I: The Role of Exceptional Divisors*, preprint `IHES/M/03/30`

[4] Brill,A., Noether,M.:*Die Entwicklung der Theorie der algebrischen Funktionen in älterer und neuerer Zeit*, Jahresber. DMV **III** (1892), pp.107-566

[5] Bodnar,G., Schicho,J.:*A Computer Program for the Resolution of Singularities*, in Resolution of Singularities (eds. H.Hauser, J.Lipman, F. Oort, A. Quiros), Progr. in Math. **181** (2000), pp. 231–238

[6] Bravo,A., Encinas,S., Villamayor,O.:*A Simplified Proof of Desingularisation and Applications*, preprint `arXiv:math.AG/0206244`

[7] Denef,J., Hoornaert,K.: *Newton Polyhedra and Igusa's local Zeta Function*, J. of Number Theory **89** (2001), pp. 31–64

[8] Gao,S.:*Factoring Multivariate Polynomials via Partial Differential Equations*, Math.Comp **72** (2003), pp. 801–822

[9] Hauser,H.:*Three Power Series Techniques*, Proc. LMS **89** (2004), pp.1–24

[10] Hironaka,H.:*Resolution of Singularities of an Algebraic Variety over a Field of Characteristic Zero*, Annals of Math. **79** (1964), pp. 109–326

[11] de Jong,A.J.:*Smoothness, Semi-Stability and Alternations*, Publ.Math.IHES **83** (1996), pp. 189–228

[12] Schulze,M., Steenbrink,J.:*Computing Hodge-Theoretic Invariants of Singularities* in D.Siersma et al.: New Developments in Singularity Theory, NATO Science Series, Kluwer (2001), pp. 235–255

[13] Greuel,G.-M., Pfister,G., Schönemann,H.: SINGULAR 2.2, to appear in 2004, `http://www.singular.uni-kl.de/`

[14] Steenbrink,J.:*Semicontinuity of the Singularity Spectrum*, Invent.Math. **79** (1985), pp. 557–565

[15] Varchenko,A.:*Asymptotic Hodge Structure in the Vanishing Cohomology*, Math. USSR Izvestija 18 (1982), pp.: 496–512

[16] Villamayor,O.:*Constructiveness of Hironaka's resolution*, Ann.Scien.Ec.Norm.Sup. 4eme serie 22 (1989), pp. 1–32

[17] Walker,R.J.:*Reduction of the Singularities of an Algebraic Surface*, Ann.Math. **36** (1935), pp. 336–365

[18] Wlodarczyk,J.:*Simple Hironaka Resolution in Characteristic Zero*, preprint `arXiv:math.AG/0401401`

[19] Zariski,O.:*The Reduction of Singularities of an Algebraic Surface*, Ann.Math. **40** (1939), pp. 639–689

[20] Zariski,O.:*A Simplified Proof for Resolution of Singularities of an Algebraic Surface*, Ann.Math. **43** (1942), pp. 583–593

Papers published in the Reports on Computer Algebra series

[RCA:33] A. Frühbis-Krüger and G. Pfister. Practical Aspects of Algorithmic Resolution of Singularities. October 2004.

[RCA:32] A. Frühbis-Krüger, K. Krüger, and H. Schönemann. Dynamic Modules in SINGULAR. December 2003.

[RCA:31] G.-M. Greuel, C. Lossen, and M. Schulze. Three Algorithms in Algebraic Geometry, Coding Theory, and Singularity Theory. March 2001.

[RCA:30] G.-M. Greuel. Applications of Computer Algebra to Algebraic Geometry, Singularity Theory and Symbolic-Numerical Solving. June 2000.

[RCA:29] G.-M. Greuel. Computer Algebra and Algebraic Geometry – Achievements and Perspectives. Feb 2000.

[RCA:28] T. Siebert. Recursive Computation of Free Resolutions and a Generalized Koszul Complex. Jan 2000.

[RCA:27] O. Bachmann and H.-G. Gräbe. The SymbolicData Project: Towards an Electronic Repository of Tools and Data for Benchmarks of Computer Algebra Software. Jan 2000.

[RCA:26] B. Reinert. Solving One-Sided Equations in Integer Monoid Rings. Jan 2000.

[RCA:25] B. Reinert and D. Zeckzer. Coset Enumeration using Prefix Gröbner Bases in Mrc – An Experimental Approach. Nov 1999.

[RCA:23] Birgit Reinert. Observations on coset enumeration. Nov 1998.

[RCA:22] K. Madlener and F. Otto. Some Applications Of Prefix-Rewriting In Monoids, Groups, And Rings. November 1998.

[RCA:21] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular version 1.2 user manual. June 1998.

[RCA:20] B. Reinert and D. Zeckzer. MRC – A System for Computing Gröbner Bases in Monoid and Group Rings. July 1998.

[RCA:19] B. Reinert, K. Madlener, and T. Mora. A note on nielsen reduction and coset enumeration. February 1998.

[RCA:18] O. Bachmann and H. Schönemann. Monomial Representations for Gröbner Bases Computations. January 1998.

[RCA:17] Thomas Siebert. An algorithm for constructing isomorphisms of modules. January 1998.